Available online at www.sciencedirect.com

SciVerse ScienceDirect

Electronic Notes in Discrete Mathematics 41 (2013) 165–172

Electronic Notes in DISCRETE MATHEMATICS

www.elsevier.com/locate/endm

An integer fixed-charge multicommodity flow (FCMF) model for train unit scheduling ⋆

Zhiyuan Lin [a,1] Raymond S. K. Kwan [a,1]

[a] *School of Computing, University of Leeds, Leeds, UK, LS2 9JT*

Abstract

An integer fixed-charge multicommodity flow (FCMF) model is used as the first part of a two-phase approach for train unit scheduling, and solved by an exact branch-and-price method. To strengthen knapsack constraints and deal with complicated scenarios arisen in the integer linear program (ILP) from the integer FCMF model, preprocessing is used by computing convex hulls of sets of points representing all possible train formations utilizing multiple unit types.

Keywords: train unit scheduling, fixed-charge multicommodity flow, convex hull

1 Introduction

A *train unit* is a set of train carriages (or *cars*) with its own built-in engine(s). Without a locomotive, it is able to move in both directions on its own. A train unit can also be coupled with other units of the same or similar types.

Given a railway operator's timetable on a particular week day, and a fleet of train units of different types, the train unit scheduling problem aims at

⋆ This research is sponsored by a Dorothy Hodgkin Scholarship funded by the Engineering and Physical Sciences Research Council (EPSRC) and Tracsis Plc.

[1] Email: {tszl,r.s.kwan}@leeds.ac.uk

http://dx.doi.org/10.1016/j.endm.2013.05.089

determining an assignment plan such that each train trip (or shortly *train*) is appropriately covered by a single or coupled units, with certain objectives achieved and certain constraints respected. From the perspective of a train unit, the scheduling process assigns a sequence of trains to it as its daily workload. The main objectives are to minimize the number of units used and/or the operational costs. It is also a common requirement to meet the passenger capacity demands.

Besides the basic requirements like trip covering, fleet size limit, compatibility between vehicle types and routes, etc, the train unit scheduling problem is found to be difficult due to some of its particular features listed below:

- Train units may be coupled/decoupled in response to passenger capacity demands for particular trains. Coupling/decoupling may take time that is not negligible, and some locations are banned/restricted for such activities.
- There are compatibility relationships among unit types when coupled.
- Unit coupling for individual trains is limited by an upper bound on the number of coupled cars, determined by many factors such as unit types, routes, platform lengths, time horizon, etc.
- Passenger capacity demand is expressed in number of seats (rather than number of units), leading to knapsack constraints that may yield very weak linear programming (LP) relaxations.
- Units may block each other on tracks and/or at platforms.

In [6], the authors have proposed a two-phase approach for the train unit scheduling problem. This approach is consisted of Phase I, an integer FCMF model, and Phase II, a multidimensional matching model resolving unit station blockage. The Phase I model is similar to [4], but important additional abilities for real-world conditions have been added, including forbidding coupling/decoupling at banned/restricted locations, ensuring coupling/decoupling time allowances, type compatibility, etc. The model explicitly uses knapsack constraints to satisfy many requirements, which has drawbacks as mentioned above. Moreover, the model uses extra variables and constraints to achieve unit type compatibility, which significantly increases the number of ILP constraints.

In this paper, we propose an updated variant of the Phase I model in [6]. The variant employs a convex hull computation preprocessing, with less number of constraints and much stronger LP relaxation, and is more flexible for complicated scenarios. The convex hull method is similar to previous works by Schrijver [8] , Ziatati et al [9], Cacchiani et al [4][3], except we deal with an improved new model with important additional features derived from [6], take a more straightforward enumeration to satisfy complicated validity rules, and consider type compatibility issues with an ad hoc branching method.

2 Model and formulation

2.1 Model description

Similar to [4] and [6], the integer FCMF model is based on a *directed acyclic graph* (DAG) $\mathcal{G} = (\mathcal{N}, \mathcal{A})$. We define the node set $\mathcal{N} = N \cup \{s, t\}$, where N is the set of *train nodes*, and s and t are the *source* and *sink* node; the arc set $\mathcal{A} = A \cup A_0$, where A is the *linkage arc* set and A_0 the *sign-on/off arc* set. A linkage arc $a \in A$ links two train nodes i and j $(a = (i, j),\ i, j \in N)$, representing a potential link such that after serving train i a unit can continue to serve train j as its next task. A sign-on arc $(s, j) \in A_0$ starts from s and ends at a train node $j \in N$; a sign-off arc $(j, t) \in A_0$ starts from a train node $j \in N$ and ends at t. Generally all train nodes have a sign-on arc and a sign-off arc. We use $\delta_-(j)$ and $\delta_+(j)$ to denote all arcs that terminate at / originate from node j respectively. Finally, a path $p \in P$ in $\mathcal{G}$ represents a sequenced daily workload (the train nodes in the path) for a unit.

For the fleet, we denote K the set of all unit types. As for type-route compatibility, type-graphs $\mathcal{G}^k$ representing routes each type $k \in K$ can serve are constructed based on $\mathcal{G}$, including all entities, e.g. P^k refers to the set of all paths in type-graph $\mathcal{G}^k$.

It is well-known that for multicommodity flow problems there are two equivalent formulations based on arcs and paths, with the same LP-relaxation bounds ([5] [4]). The model in this paper uses the latter, with *path variable* $x_p \in \mathbb{Z}_+, \forall p \in P^k, \forall k \in K$ to indicate the number of units used in path p. Note there is other *blockflow variable* $y_a \in \{0, 1\}, \forall a \in \mathcal{A}$ indicating whether an arc a is used.

2.2 Model ILP formulation

$$(P) \quad \min \quad W_1 \sum_{k \in K} \sum_{p \in P^k} c_p x_p + W_2 \sum_{a \in \mathcal{A}} y_a \tag{1}$$

$$\sum_{p \in P^k} x_p \leq b^k, \quad \forall k \in K; \tag{2}$$

$$\sum_{k\in K_j}\sum_{p\in P_j^k} D_{f,k}^j x_p \le d_f^j, \quad \forall f \in \bar{F}_j, \forall j \in N. \tag{3}$$

$$\sum_{k\in K_a}\sum_{p\in P_a^k} x_p \le m_a y_a, \quad \forall a \in \mathcal{A}; \tag{4}$$

$$\sum_{a\in\delta_-(j)} y_a = 1,\ \forall j \in N_B^-; \quad \sum_{a\in\delta_+(j)} y_a = 1,\ \forall j \in N_B^+; \tag{5}$$

$$\tau^D_{\text{arr}(i)}\left(\sum_{a\in\delta_+(i)} y_a - 1\right) + \tau^C_{\text{dep}(j)}\left(\sum_{a\in\delta_-(j)} y_a - 1\right) \le e_{ij}, \quad \forall (i,j) \in A^*; \tag{6}$$

$$x_p \in \mathbb{Z}_+,\ \forall p \in P^k, \forall k \in K; \quad y_a \in \{0,1\},\ \forall a \in \mathcal{A}. \tag{7}$$

In the first term of Objective (1), c_p is the overall cost for path p. It can be a weighted linear combination of several sub-costs to achieve multiple effects and/or preferences. The second term minimizes total number of *used* arcs, leading to minimization of the total number of coupling/decoupling activities, thus eliminating unnecessary ones; it also partly drives y_a to desired binary values. $W_{1,2}$ are weights for the two terms, and generally $W_1 > W_2$.

Constraints (2) ensure the deployed number of units for each type $k \in K$ is within its fleet size upper bound b_k. Constraints (3) are used to fulfil the following requirements for individual trains: (i) passenger capacity demands; (ii) coupling upper bounds due to various factors; and (iii) type compatibility. Constraints (3) will be discussed in detail in Section 2.3. Constraints (4) calculate binary blockflow variables $y_a, \forall a \in \mathcal{A}$ such that $y_a = \begin{cases} 1, & \sum_{k\in K_a}\sum_{p\in P_a^k} x_p > 0, \\ 0, & \sum_{k\in K_a}\sum_{p\in P_a^k} x_p = 0, \end{cases}$ where P_a^k is the set of paths in $\mathcal{G}^k$ passing through arc a, K_a is the type set allowed at arc a, and m_a is the maximum number of coupled units flowing through arc a. Constraints (5) are to forbid coupling/decoupling at banned locations, where $N_B^- \subseteq N$ is the set of trains whose departure locations are banned for coupling/decoupling, and $N_B^+ \subseteq N$ for arrival locations. Constraints (6) are to ensure time allowance validity for coupling/decoupling at some linkage arcs $A^* \subseteq A$ where a time violation may occur; $\tau^D_{\text{arr}(i)}$ is the time for a single decoupling operation at the arrival location of train i, $\tau^C_{\text{dep}(j)}$ the time for a single coupling operation at the departure location of train j, and e_{ij} is the time slack between i and j. Finally (7) gives the variable domain.

2.3 Computation for convex hulls associated with trains

Similar to [4], let K_j be the set of permitted types for train $j \in N$, and $w^j = (w_1^j, w_2^j, \cdots, w_{|K_j|}^j)^T \in \mathbb{Z}_+^{K_j}$, where w_k^j is the number of units of type k used for j. We define a *unit combination set* by enumerating all valid unit formation for a train:

$$\mathcal{W}_j := \left\{ w^j \in \mathbb{Z}_+^{K_j} \middle| \forall w^j : \text{a valid unit combination for train } j \right\}, \tag{8}$$

such that

(i) $\sum_{k \in K_j} q_k w_k^j \geq r_j$, where q_k is the capacity of unit type k and r_j is the passenger capacity demand, both measured in numbers of seats.

(ii) $n(w^j) \leq u(w^j), \forall w^j \in \mathcal{W}_j$, where $n(w^j)$ and $u(w^j)$ are number of coupled cars and coupling upper bound for combination w^j respectively.

(iii) the used types $(k : w_k^j > 0)$ are compatible;

Notice while it is easy to satisfy demand requirement as in (i) by linear constraints, it is generally very difficult for (ii) and (iii) in the same way. Moreover, enumeration generally suits any validity rules, which in our cases are more complicated than those in [4] and [6]. It also avoids knapsack constraints and excludes a large proportion of incompatible types.

For all $\mathcal{W}_j$ in our instances, due to their small dimensions ($|K_j| \leq 4, \forall j \in N$), and numbers of points ($|\mathcal{W}_j| \leq 9, \forall j \in N$), it is possible to explicitly compute the convex hulls of all unit combination sets, which we refer to as *train convex hulls*:

$$\operatorname{conv}(\mathcal{W}_j) = \left\{ w^j \in \mathbb{R}_+^{K_j} \middle| D^j w^j \leq d^j \right\}, \quad \forall j \in N, \tag{9}$$

described by a set of facets $f \in F_j$, with $D^j \in \mathbb{R}^{F_j \times K_j}$ and $d \in \mathbb{R}^{F_j}$. Note we only need the nonzero facets $f \in \bar{F}_j \subseteq F_j$. Finally, all w-variables will be replaced by x-variables in the same way as in [4], which forms Constraints (3) in (P).

Example We use an example from the real data of a UK rail operator to illustrate the above preprocessing approach. For a train "1C11", train units of Type 318 (3-car, 219 seats), 320 (3-car, 230 seats) and 156 (2-car, 145 seats) are permitted, and only Type 318 and 320 are compatible for coupling. When served by Type 318 and/or 320, the coupling upper bound is 6 cars, and when served by Type 156, this bound changes to 4 cars. In addition, "1C11" has a passenger capacity demand of 230 seats. Then we have:

$$\mathcal{W}_{1C11} = \{(w_{318}, w_{320}, w_{156}) | (2,0,0), (0,1,0), (0,2,0), (1,1,0), (0,0,2)\},$$

and its corresponding train convex hull:

$$\text{conv}(\mathcal{W}_{1C11}) = \left\{ w \in \mathbb{R}^3_+ \;\middle|\; \begin{array}{l} f_1 : w_{318} + 2w_{320} + w_{156} \geq 2 \\ f_2 : w_{318} + w_{320} + w_{156} \leq 2 \end{array} \right\},$$

which is a polytope with two nonzero facets $\{f_1, f_2\} = \bar{F}_{1C11}$, giving two corresponding constraints for "1C11" in Constraints (3).

3 Solution approach

3.1 A branch-and-price ILP solver

The above ILP (P) is solved by a branch-and-price [2] method based on column generation [7] where path variables x_p are generated dynamically.

Let $\alpha^k \leq 0, \beta_{f,j} \leq 0, \gamma_a \leq 0$ be the dual variables from Constraints (2), (3) and (4), N_p and $\mathcal{A}_p$ the set of train nodes and arcs in path p respectively, the reduced cost of a path $p \in P^k$ is,

$$\bar{c}_p = c_p - \alpha^k - \sum_{j \in N_p} \sum_{f \in \bar{F}_j} D^j_{f,k} \beta_{f,j} - \sum_{a \in \mathcal{A}_p} \gamma_a, \tag{10}$$

finding the smallest value of which can be regarded as a shortest path problem with train node weight $-\sum_{f \in \bar{F}_j} D^j_{f,k} \beta_{f,j}, \forall j \in N^k$, arc weight $-\gamma_a, \forall a \in \mathcal{A}^k$, plus a source-sink weight $c_p - \alpha^k$. There are $|K|$ subproblems from Dantzig-Wolfe decomposition.

3.2 Branching rules

Although all vertices are feasible points for each $\text{conv}(\mathcal{W}_j)$, it may still contain points with incompatible types serving the same train, which have to be eliminated. This can be realized by a *train-family branching* proposed in [6]. We introduce the concept of *train unit family* such that unit types that are compatible belong to the same family. The symbol Φ_j is used to denote the set of all families that can serve train j.

The main idea of train-family branching is to check the LP relaxation solution and select a train j that is covered by more than one family, say families $\phi_1, \phi_2, \cdots, \phi_n$ (n is usually not a large number). Then we form $n+1$ branches with respect to families $\phi_1, \phi_2, \cdots, \phi_n$.

- For the first n branches $1, \cdots, n$, say at a branch $i \in \{1, \cdots, n\}$, only family ϕ_i is allowed to serve train j. To achieve this, in the restricted master problem (RMP), all paths indicating any families in $\Phi_j \setminus \{\phi_i\}$ serving j are deleted; in the shortest path problem of type k whose family is not ϕ_i, node j is deleted from the shortest path network.
- In the last $(n+1)$th branch, if $|\Phi_j \setminus \{\phi_1, \cdots, \phi_n\}| \geq 1$, then we forbid families $\phi_1, \cdots, \phi_n$ to serve train j, which can be realized by similar path/node deleting ways as described above; if $\Phi_j = \{\phi_1, \cdots, \phi_n\}$, then the $(n+1)$th branch is no longer needed.

Notice this branching rule does not add any extra constraints to the RMP. It actually reduces the number of columns in the RMP and the network scale of the shortest path subproblems, which effectively divides the search space and forces the trains to be served by compatible types.

It should be mentioned that it is not enough to drive all variables into integers only by train-family branching. When all trains have been served by compatible types, the branching strategy will be switched to *arc variable branching*, e.g. the method proposed in [1] for integer multicommodity flow problems.

4 Preliminary experiments and conclusions

The experiments are based on the data from a UK rail operator with around 2250 train trips on a weekday. All possible unit combinations have been computed, and the `convhull()` and `convhulln()` functions provided by Matlab R2012a are used for convex hull computation. We find that the numbers of nonzero facets of train convex hulls are very small (2.11 facets per train on average), which can be proved in a similar way as in [9]. Among the 2250 trains, 51 trains (2.2%) have a number of nonzero facets of 1, 1987 trains (88.3%) of 2, 176 trains (7.8%) of 3, 13 trains (0.57%) of 5, and 23 trains (1.0%) of 7. This implies in the ILP (P) the number of constraints will be greatly reduced, compared with the original model. Further results and analysis will be presented at INOC2013 and in a post-conference full paper.

This paper has proposed an updated model of a previous version for real-world train unit scheduling. A method by explicitly computing convex hulls for each train is used thereby making it easier to solve the complex integer model within practical computational time. The preliminary experiments have shown promising results that the numbers of facets for most train convex hulls are small.

Testing and analyzing the final scheduling results with the rail operator we are collaborating are ongoing.

References

[1] Alvelos, F., "Branch-and-Price and Multicommodity Flows," Ph.D. thesis, Escola de Engenharia, Universidade do Minho, Portugal (2005).

[2] Barnhart, C., E. L. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh and P. H. Vance, *Branch-and-price: Column generation for solving huge integer programs*, Operations Research **46** (1998), pp. 316–329.

[3] Cacchiani, V., A. Caprara, G. Maroti and P. Toth, *On integer polytopes with few nonzero vertices*, Operations Research Letters **41** (2013), pp. 74–77.

[4] Cacchiani, V., A. Caprara and P. Toth, *Solving a real-world train-unit assignment problem*, Mathethatical Programming **124** (2010), pp. 207–231.

[5] Cook, W. J., W. H. Cunningham, W. R. Pulleyblank and A. Schrijver, "Combinatorial Optimization," Wiley, New York, 1998.

[6] Lin, Z. and R. S. K. Kwan, *A two-phase approach for real-world train unit scheduling*, in: *Proceedings of the 12th Conference on Advanced Systems for Public Transport*, Santiago, Chile, 2012.

[7] Lübbecke, M. E. and J. Desrosiers, *Selected topics in column generation*, Operations Research **53** (2002), pp. 1007–1023.

[8] Schrijver, A., *Minimum circulation of railway stock*, CWI Quarterly **6** (1993), pp. 205–217.

[9] Ziarati, K., F. Soumis, J. Desrosiers and M. M. Solomon, *A branch-first, cut-second approach for locomotive assignment*, Manage Sci **45** (1999), pp. 1156–1168.