Modelling of complex costs and rules in a crew pairing column generator

Rastislav Galia and Curt Hjörring

Carmen Systems AB, Odinsgatan 9, Sweden
rastislav.galia@carmensystems.com
curt.hjorring@carmensystems.com

Abstract. Crew pairing, the creation of anonymous lines of work, is a crucial part of the crew airline planning process. Column generation with shortest path pricing subproblem provides high quality solutions. In its basic form, the pricing subproblem relies on assumptions, such as additivity of the cost function and constraint contributions. However, it is not possible to be assume that these requirements are satisfied, particularly if the pairing system gives the user control over problem formulation and maintenance.

Solutions to these challenges are proposed, based on proper granularity of the subproblem, a k–shortest paths based pricer and the application of resources to model nonadditive costs. Furthermore, a label–merging technique provides significant performance improvements.

1 The pairing problem

The pairing problem consists of creating anonymous sequences of flight legs (pairings), so that the coverage requirement of each leg is fulfilled. Each pairing is required to start and end at the same homebase airport and satisfy legality constraints (rules). The rules are often of a complex structure and vary from airline to airline.

Pairing costs consists mainly of salary, per diem compensation, hotel costs and artificial penalties introduced to reward some characteristics of the pairings. The resulting cost function is in general nonadditive. The optimisation problem is to minimise the sum of all the pairing costs.

1.1 Approaches survey

The pairing problem can be formulated as a set–covering problem. Solving this problem to optimality requires explicit enumeration of all legal pairings, which is computationally very expensive due to the very large number of legal pairings.

The problem can instead be solved approximately by an iterative approach. In each iteration the previous solution is used as a starting point. New pairings that are similar to the pairings in the previous solution are generated and the iteration is concluded by solving a set–covering IP problem with pairings from the previous solution and the newly generated ones.

Various heuristic strategies are used in the generation step, for example time–windowing. This is known as the *generate and optimise* approach. Further details can be found in [2].

Another widely used technique, column generation, is based on the idea of solving the LP–relaxation of the original problem. New pairings with negative reduced costs are generated by solving the pricing subproblem. If the *pricing subproblem* is solved optimally, this method will find the optimal solution to the LP–relaxation.

An early application of column generation to the pairing problem is due to [7], in which it is shown that columns can be priced by solving a shortest path problem on a network with arcs representing legs and overnight connections. In the airline industry the sequence of legs flown in a working day is known as a *duty period*, and for many rules the legality of a single duty period is independent of preceding or succeeding duties. This additional structure is exploited in [5] where a duty period network is formed in which nodes are duty periods with state information, and arcs represent legal overnights.

Results for flight and duty based implementations are presented in [8]. In both versions, resources (labels) are added to each node to track legality conditions, and a *resource constrained shortest path problem* is solved. Comparisons between the two versions show that flight based implementation generally spends more time in the pricing routine. However, the duty version cannot solve as large problems as the flight version due to memory limitations.

The problem of the size of the duty network is addressed in [4]. A relaxed duty network with a significantly smaller number of arcs is introduced. This has a positive impact on memory consumption as well as the time spent in the pricing routine. A refinement scheme for the duty network (partial reversing of the relaxation) is proposed.

In the next sections we will build on top of the framework proposed by [4] by extending the class of modelled rules and cost functions. In section 2 we show how some results can be achieved by the proper design of the network topology. Some more complex rules and costs require modification of the pricing subproblem solver, as shown in section 3.

2 The solution method

2.1 Rules and costs, their structure and evaluation

Carmen Systems provides a rule modelling language (Rave) that permits planners to easily add and modify the rules and the cost function. Rave is a black-box system, able to evaluate expressions on chained legs and deciding whether a chain is legal with respect to the rules. However, it does not expose the internal details of the evaluation process to the rest of the system.

The structure of the rules and costs has significant impact on the pricing process. For modelling purposes the sequence of legs is usually grouped into

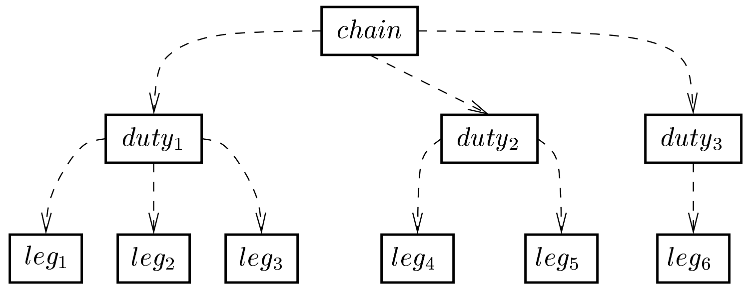

Fig. 1. Hierarchy of chain levels

the hierarchy of levels, as shown in the figure 1. In the pairing problem, there is usually only one intermediate level of *duties* between the atomic level of the legs and the top–most level.

An important concept is the range of objects that evaluated expressions depend on. For example, an expression returning "block time of current duty" has range of one duty. An expression determining the length of the pairing in terms of days has chain range.

2.2 Network model

The pricing routine, as proposed by [7], finds a pairing with negative reduced cost as a shortest path in the network. This basic approach relies on the additivity of the cost function and the legality of the pairings corresponding to the path in the network.

These assumptions are very seldom satisfied entirely, so [4] proposes to introduce a k–shortest path (k–SP) based pricing subproblem solver. In this design, the pricing routine works by first calculating the shortest path tree for the network. This returns the cheapest path in the network. If the cost is nonnegative, there are no attractive pairings, and the pricing routine terminates.

If the path has a negative reduced cost estimate (which is obtained by summing up the arc costs), the rule system is called to see if the corresponding pairing is legal, and if it also has negative reduced cost. If so the pairing is added to the LP, and the pricing routine can either terminate, or continue producing attractive pairings. If the pairing is illegal, or does not have negative reduced cost, then the pricing routine finds the next shortest path, and repeats the tests. This continues until one of the above termination criteria is satisfied, or all paths are enumerated.

As mentioned in [4] and [5], the fact that the majority of the pairing rules have range of one or two duties can be exploited to design an efficient pricing routine. Approximate additivity of the cost function with respect to the duties and duty–duty connections reduces the frequency and magnitude of cost estimation errors as well.

2.3 Network topology

The network is designed to model completeness criteria (a pairing has to start and end at the same base), maximum length of the pairing in calendar days and nonadditive penalty dependent on the length of the pairing.

This is illustrated in figure 2. There are multiple source and sink nodes, one for each calendar day and homebase combination. A separate k–SP pricing is performed for each starting node, and the search is restricted to paths leading to the corresponding homebase within a specified number of days. This way the maximum length of the pairing and the completeness rule is satisfied. Arcs between the sink nodes and the "master sink" node model calendar length penalty.

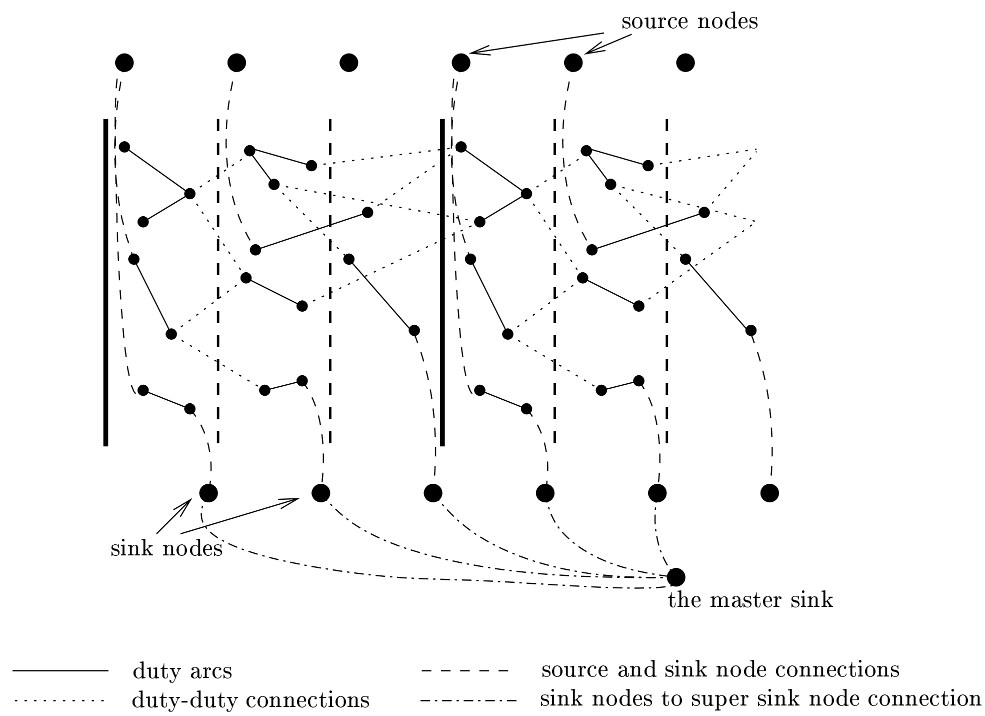

Fig. 2. Overview of the network model

3 Extension to the network model

The basic network model gives the foundation for a successful combination of column generation with a black–box rule system. However, for some classes of problems the cost estimation errors and the rule failures can severely impact the performance or the solution quality. Here we present how one such class of problems can be modelled in the network and how this change affects the k–SP routine.

3.1 Resource dependent cost elements

The resource constrained shortest path problem has traditionally been used as a subproblem in applications of column generation to scheduling problems. The technique can be extended to model some classes of nonadditive costs as well.

Let us assume that cost of a pairing can be written in the form $c(p) = c_a(p) + c_n(r(p))$, where $c_a(p)$ is additive with respect to the duties and duty connections. $c_n(r(p))$ is the nonadditive part dependent on additive resource vector $r(p)$.

Not every nonadditive cost can be modelled using resources. The resource must be a additive function. Furthermore, the nonadditive part of the cost c_n is required to be nonnegative and nondecreasing with respect to the value of the individual resources.

In practise, c_n may depend also on some other characteristics of the path that can be determined directly from the network, such as starting day of the pairing (given by path's source node) or the length in calendar days. Some examples of costs modelled by resources:

- Penalise pairings for which the flight time exceeds 480 minutes by constant value of 4000 cost units.
- Penalise pairings for number of the short night stops ($<$ 6 hours). One short night–stop is not penalised, two or more short night–stops are penalised by 500.

Modification of the network and the k–SP algorithm for use with the resources

The network model is modified in such a way that additive part $\bar{c}_a$ of the reduced cost is kept on the arcs instead of the original reduced cost $\bar{c}$. In addition, arc *resource consumption* vector $r(p)$ is stored in the graph as well.

The pricing subproblem consists of finding any attractive pairing (with negative reduced cost $\bar{c}(p) = c(p) - \pi A$). However, it is preferable to find several attractive pairings in one pricing iteration.

If there are no resources, the k–SP algorithm achieves this by enumerating all paths in ascending order with respect to the reduced cost. It stops after encountering the first non-attractive path, at which point all attractive paths have been found.

With resources, a nonadditive pricing subproblem needs to be solved and the k–SP algorithm can not guarantee that the generated paths are ordered by their reduced cost, as illustrated by the example in figure 3. In the example pricing subproblem the cost function is the sum of a additive cost and a nonadditive part c_n depending on one resource. Let us assume, that $c_n(r(p)) = 0$ if $r(p) < 70$ and $c_n(r(p)) = -5$ otherwise.

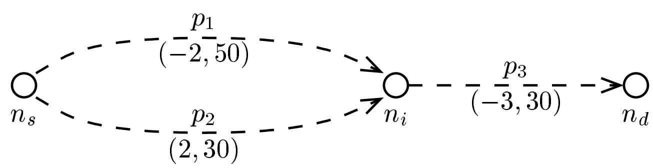

Fig. 3. Impact of the resources on the k–SP algorithm

Even though the cost of the path p_1 is $-2 + c_n(50) = -2$ and cost of the path p_2 is $2 + c_n(30) = 2$, we cannot say that p_1 is definitely better than p_2, since $\bar{c}(p_1 \circ p_3) = -5 + c_n(80) = 0 > \bar{c}(p_2 \circ p_3) = -1 + c_n(60) = -1$.

Attractive paths could be found by enumerating all paths from the source to the sink node. However, this process is obviously very inefficient. In order to reduce the number of generated paths, we search for the set of *Pareto–optimal* paths from the source to the sink node.

Definition 1. The path p is said to be *dominated* by path p', if $\bar{c}_a(p) \geq \bar{c}_a(p')$ and $r_j(p) \geq r_j(p')$ for every resource r_j.

Definition 2. Let $P(n_s, n)$ be set of all paths from the source node to node n. A path is called *Pareto–optimal*, if it is not dominated by any path in $P(n_s, n)$.

If the set $P(n_s, n_d)$ of paths from the source node n_s to the sink node n_d contains at least one attractive path, then it contains also at least one attractive Pareto–optimal path. This means that the pricing subproblem is guaranteed to be solved by this approach.

A modification of the algorithm due to [3] is used to find the set of the Pareto–optimal paths. An important property of the Pareto–optimality is the “principle of optimality”, meaning that the subpath to an intermediate node in the network, which is not Pareto–optimal, cannot be part of a Pareto–optimal complete path to the sink node. This allows us to discard dominated subpaths in the early stages of the network and to significantly reduce the computational effort.

Pareto–optimal subpaths to each node are enumerated in increasing lexicographic order with respect to $\bar{c}_a$ and vector of resources $r(p)$. The k–SP terminates after encountering the first path to sink node with $\bar{c}_a \geq 0$.

Label merging

Due to frequent rule failures and underestimates, the k–SP routine can spend very long time in certain parts of the network until it finds the attractive pairings. These problems might be resolved in later stages thanks to a refinement of the duty network. However, it is preferable to focus the search effort to those parts of the network, which are more likely to produce attractive legal rotations.

A simple adaptive technique can be used to regulate the number of generated non–dominated paths in the network by over–restricting the dominance with the obejctive to dominate a larger number of paths which are represented by a "merged label" (path).

Path p is said to be dominated by path p' if $\bar{c}_a(p) + \kappa_a(n_s) \geq \bar{c}_a(p')$ and $r^{(j)}(p) + \kappa^{(j)}(n_s) \geq r^{(j)}(p')$ for each resource $r^{(j)}$, where $\kappa_a(n_s), \kappa^{(j)}(n_s)$ are parameters depending on the current source node n_s. Parameters $\kappa_a(n_s)$ and $\kappa^{(j)}(n_s)$ are adaptively changed after every pricing iteration, depending on the time spent in a pricing routine for source node n_s.

4 Results

The extensions to the network model and pricing subproblem solver mentioned in section 3 were successfully used in a commercial production system developed by Carmen Systems.

The improved modelling capabilities have been tested on several problem instances of our clients and we observed significant performance improvements. Figure 4 depicts the effect of using the resource modelling and label merging on a pairing problem from KLM. The graph shows that using both new features in the pricer can reduce runtimes by up to 80% on some examples.

5 Summary

By extending the duty–network approach in [4] we have developed a flexible column generator system able to handle wide variety of cost and rule structures while maintaining the black-box paradigm of flexible rule modelling language. In order to better support this flexibility, more sophisticated network algorithms employing resource modelling and label merging were used for solving the pricing subproblem.

References

1. R. Anbil, J. J. Forrest, and W. R. Pulleyblank. Column generation and the airline crew pairing problem. *Documenta Mathematica*, Extra Volume ICM III:677–686, 1998.
2. E. Andersson, E. Housos, N. Kohl, and D. Wedelin. Crew pairing optimization. In G. Yu, editor, *Operations Research in Airline Industry*, pages 228–258. Kluwer Academic Publishers, Norwell, MA, 1998.

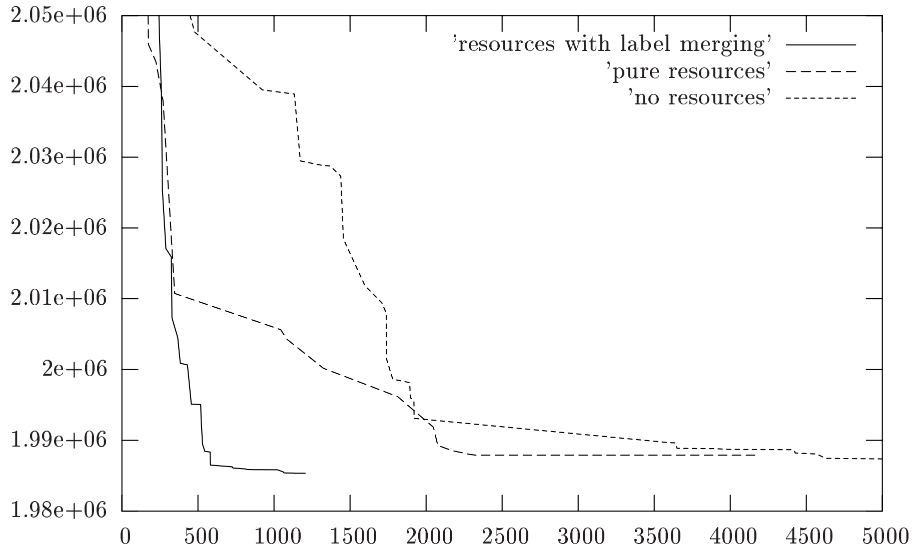

Fig. 4. Behaviour of column generator resource modelling

3. J. A. Azevedo, E. Q. V. Martins. An algorithm for the multiobjective shortes path problem on acyclic networks. *Investigação Operacional*, Vol. 11, pages 52–69, 1991.
4. C. Hjorring and J. Hansen. Column generation with a rule modelling language. In *Proceedings of the 34th Annual Conference of the Operational Research Society of New Zealand*, pages 133–142, Hamilton, New Zealand, December 10–11, 1999.
5. S. Lavoie, M. Minoux, and E. Odier. A new approach of crew pairing problems by column generation and application to air transport. *European Journal of Operations Research*, pages 45–58, 1988.
6. E. Martins and J. Santos. *A new shortest paths ranking algorithm*. Investigação Operacional, 20(1):47-62, 2000.
7. M. Minoux. Column generation techniques in combinatorial optimization: A new application to crew pairing problems. In *XXIVth AGIFORS Symposium*, 1984.
8. P. H. Vance, A. Atamtürk, C. Barnhart, E. Gelman, E. L. Johnson, A. Krishan, D. Mahidhara, G. L. Newhauser, and R. Rebello. A heuristic branch-and-price approach for the airline crew pairing problem. Technical report, School of Industrial and Systems Engineering, Georgia Tech., June 1997.