Airline Crew Scheduling by Column Generation

Guy Desaulniers and François Soumis

École Polytechnique, Canada

Spring School on Combinatorial Optimization in Logistics
Montréal, May 17-20, 2010

Collaborators

- Khaled Boubaker, MSc student
- Issmail Elhallaoui, PhD student and post-doc
- Alain Hertz, professor
- Abdelmoutalib Metrane, post-doc
- Mohammed Saddoune, PhD student
- Personnel from Ad Opt/Kronos

Outline

1. Airline planning process
2. Crew pairing and column generation
3. Crew assignment and dynamic constraint aggregation
4. Integrated crew scheduling and bi-dynamic constraint aggregation
5. Conclusions and future work

Outline

Planning process

- Operations research provides several tools to the airlines for planning their operations
- This planning process aims at maximizing the airline profits
- However, it is a complex process that involves several departments:
  - Marketing for flight scheduling
  - Scheduling for aircraft assignment and routing
  - Flight operations for crew scheduling

Planning process

- Operations research provides several tools to the airlines for planning their operations
- This planning process aims at maximizing the airline profits

- However, it is a complex process that involves several departments:
  - Marketing for flight scheduling
  - Scheduling for aircraft assignment and routing
  - Flight operations for crew scheduling

To simplify it, this process is typically divided into four main steps:

1 - Flight scheduling

- Given aircraft availability per type, determine the schedule of flights to operate over a given season in order to maximize anticipated profits
- Cyclic weekly problem with exceptions for holidays
- Starts usually from the previous year schedule
- Requires a passenger flow model to evaluate revenues
    - Estimated passenger demand per day and OD
    - Passenger preferences (departure time, number of legs, etc.)
    - Seating capacity per flight
    - Output: number of passengers per itinerary

2 - Fleet assignment

- Given a weekly flight schedule, aircraft availability per type, estimated profits for each flight and aircraft type, find the aircraft type to assign to each flight so as to maximize total estimated profits while ensuring aircraft flow conservation in the network
- Cyclic weekly problem with exceptions for holidays
- Profits per flight and type are approximated via a passenger flow model (iterative process)

3 - Aircraft routing

- Given a flight schedule for an aircraft type and aircraft availability for this type, determine aircraft routes that satisfy short-term maintenance requirements
- One maintenance (inspection lasting about 4 hours) at most at every three or four days
- Cyclic weekly problem or dated monthly problem
- In general, feasibility problem or through value maximization problem

4 - Crew scheduling

- Given a monthly flight schedule for an aircraft type, determine the schedules of the crew members
- Typically divided into two steps: crew pairing and crew assignment

Remarks

- This decomposition process is heuristic and often requires backtracking
- For small- and medium-sized airlines, certain steps can be integrated

4 - Crew scheduling

- Given a monthly flight schedule for an aircraft type, determine the schedules of the crew members
- Typically divided into two steps: crew pairing and crew assignment

Remarks

- This decomposition process is heuristic and often requires backtracking
- For small- and medium-sized airlines, certain steps can be integrated

Outline

Crew pairing problem

Separable by

- Crew category: cockpit (pilots and copilots) and cabin (flight directors and attendants)
- Aircraft type or family of types

Definitions

- Crews are assigned to a few crew bases
- A pairing is a sequence of duties interspersed with rest periods, starting and ending at a crew base
- A duty is a sequence of flights, deadheads and connections forming a working day
- A deadhead is a flight (of any company) on which the crew travels as passengers

Crew pairing problem

Separable by

- Crew category: cockpit (pilots and copilots) and cabin (flight directors and attendants)
- Aircraft type or family of types

Definitions

- Crews are assigned to a few crew bases
- A pairing is a sequence of duties interspersed with rest periods, starting and ending at a crew base
- A duty is a sequence of flights, deadheads and connections forming a working day
- A deadhead is a flight (of any company) on which the crew travels as passengers

Cost

The cost of a pairing is a complex function of:

- flying time per duty (minimum guaranteed)
- span of each duty
- span of the pairing
- deadhead costs
- accomodation fees (for rests outside the base)

Feasibility

A pairing is feasible if it satisfies all safety and collective agreement rules such as:

- maximum number of calendar days in a pairing
- maximum number of duties in a pairing
- minimum rest time between two consecutive duties
- maximum number of landings per duty
- maximum span of a duty
- maximum flying time per duty
- minimum connection time between two consecutive flights

Crew pairing problem statement

- Given
  - a set of scheduled flights
  - safety and working rules
  - minimum or maximum credited hours per crew base
- Find least-cost feasible crew pairings
- Such that
  - each flight is covered by an active crew
  - the minimum or maximum credited hours per crew base is respected

Remark

A typical crew pairing problem instance is defined for a one-month dated flight schedule and solved one month ahead of time

Crew pairing problem statement

- Given
  - a set of scheduled flights
  - safety and working rules
  - minimum or maximum credited hours per crew base
- Find least-cost feasible crew pairings
- Such that
  - each flight is covered by an active crew
  - the minimum or maximum credited hours per crew base is respected

Remark

A typical crew pairing problem instance is defined for a one-month dated flight schedule and solved one month ahead of time

Traditional sequential approach

- Solve a cyclic daily problem
- Unfold the computed solution over a typical week (one copy of each pairing for each week day) and remove infeasible copies
- Solve a cyclic weekly problem preserving as much as possible the unfolded daily solution
- Unfold the computed weekly solution over the month (one copy of each pairing for each week of the month) and remove infeasible copies
- Solve the dated monthly problem preserving as much as possible the unfolded weekly solution

Set partitioning type model

$$\text{Minimize} \quad \sum_{p \in P} c_p y_p \tag{1}$$

$$\text{s.t.} \quad \sum_{p \in P} a_{fp} y_p = 1, \quad \forall f \in F \tag{2}$$

$$\sum_{p \in P} b_{qp} y_p = e_q, \quad \forall q \in Q \tag{3}$$

$$y_p \in \{0, 1\}, \quad \forall p \in P \tag{4}$$

P: set of all feasible pairings;

c_p: cost of pairing $p \in P$;

F: set of flights to cover;

a_{fp}: binary parameter taking value 1 if flight $f \in F$ is actively covered in pairing $p \in P$ and 0 otherwise;

Q: set of indices for the side constraints;

b_{qp}: contribution of pairing $p \in P$ to side constraint $q \in Q$;

e_q: right-hand side of side constraint $q \in Q$;

y_p: binary variable taking value 1 if pairing p is selected and 0 otherwise.

- **A priori generation of a subset of pairings**
  - Anbil *et al. (1992), Hoffman and Padberg (1993), Chu et al. (1997)*
- **Dynamic column generation based on resource-constrained shortest path subproblems**
  - Desaulniers *et al. (1997), Vance et al. (1997), Barnhart and Shenoi (1998)*
- **Dynamic column generation based on truncated depth-first search enumeration**
  - Marsten (1994), Anbil *et al. (1998), Klabjan et al. (2001), Makri and Klabjan (2004)*
- **Robust crew scheduling**
  - Erghott and Ryan (2002), Chebalov and Klabjan (2006)

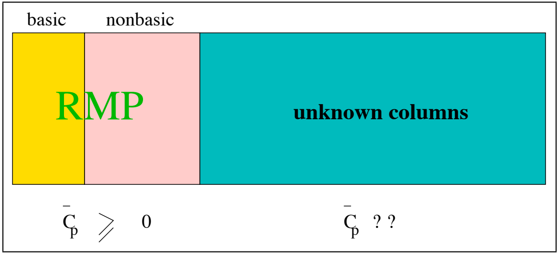
Column generation

Basics

- Column generation is used to solve the linear relaxation of model (1)-(4), which is called the master problem
- Iterative method alternating between a restricted master problem (RMP) and several subproblems
- At iteration i, the RMP is simply the master problem restricted to a subset of its variables
- The RMP is solved by a linear programming solver to produce a primal and a dual solution

Basics (cont'd)

- This primal solution is optimal for the master problem if the reduced costs of all variables are nonnegative
- This condition holds for all known variables appearing in the RMP

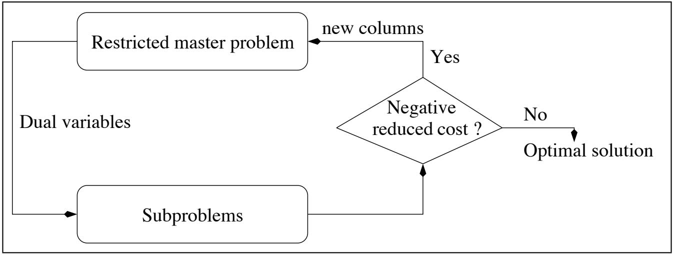


Basics (cont'd)

- One or several subproblems must be solved, using a specialized algorithm, to determine if it holds for the unknown variables
- Each subproblem searches for a least reduced cost variable among a subset of variables
- If the least reduced cost is nonnegative for all subproblems, then the primal solution of the current RMP is also optimal for the master problem and the column generation algorithm stops
- Otherwise, negative reduced cost variables identified by the subproblems are added to the RMP before starting a new iteration

Basics (cont'd)

- One or several subproblems must be solved, using a specialized algorithm, to determine if it holds for the unknown variables
- Each subproblem searches for a least reduced cost variable among a subset of variables
- If the least reduced cost is nonnegative for all subproblems, then the primal solution of the current RMP is also optimal for the master problem and the column generation algorithm stops
- Otherwise, negative reduced cost variables identified by the subproblems are added to the RMP before starting a new iteration



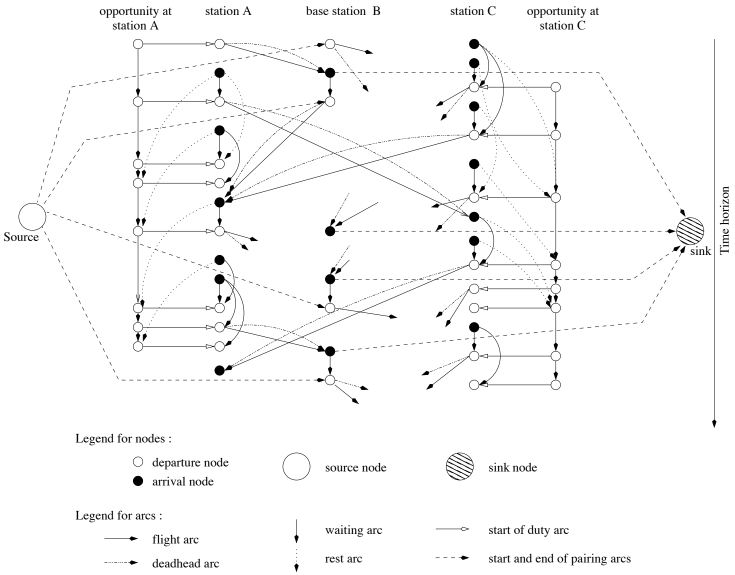
Starts by solving the RMP with

- Artificial variables or
- Initial set of columns containing a feasible RMP solution

Subproblems

For the weekly crew pairing problem

- There is one subproblem per crew base and per day of the week
- Such a subproblem is a shortest path problem with resource constraints that aims at finding a feasible pairing starting on the corresponding day at the corresponding base with the least reduced cost
- A subproblem is defined over an acyclic time-space network


opportunity at station A
station A
base station B
station C
opportunity at station C
Source
sink
Time horizon
Legend for nodes :
departure node
arrival node
source node
sink node
Legend for arcs :
flight arc
deadhead arc
waiting arc
rest arc
start of duty arc
start and end of pairing arcs

Path feasibility

- Each feasible pairing starting at this base and on this day is represented by a source-to-sink path in this network
- However, not all paths correspond to a feasible pairing
- Resource constraints are used to restrict path feasibility
- A resource is a quantity that varies along a path and its value must fall within a prespecified resource window (interval) at each node

Path feasibility

- Each feasible pairing starting at this base and on this day is represented by a source-to-sink path in this network
- However, not all paths correspond to a feasible pairing

- Resource constraints are used to restrict path feasibility
- A resource is a quantity that varies along a path and its value must fall within a prespecified resource window (interval) at each node

Resource constraints: example

A resource can be used to limit the number of landings per duty (maximum 5)

- The resource window is [0,0] at the source node and [0,5] at every other node
- The resource increases by 1 on every flight and deadhead arc
- It is reset to 0 on every rest arc

Remark

For a real-life crew pairing problem, between 5 and 10 resources are needed

Resource constraints: example

A resource can be used to limit the number of landings per duty (maximum 5)

- The resource window is [0,0] at the source node and [0,5] at every other node
- The resource increases by 1 on every flight and deadhead arc
- It is reset to 0 on every rest arc

Remark

For a real-life crew pairing problem, between 5 and 10 resources are needed

Arc (reduced) costs

- The cost of a feasible path p must be equal to the reduced cost of the corresponding variable y_p
- Every arc bears its original cost, possibly modified by dual values
- The dual variable π_f associated with (2) for flight f is subtracted on every flight arc representing f
- The dual variables associated with (3) must also be subtracted
  - For instance, if a constraint (3) limits the number of credited hours for a base w, then $b_a\sigma_w$ must be subtracted on an arc a in a subproblem for base w where b_a is the number of credited hours allocated on arc a and σ_w the corresponding dual variable

The shortest path subproblems with resource constraints are solved by a labeling algorithm

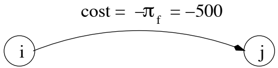
- A label represents a partial path starting at the source node
- A label contains one component per resource and one (reduced) cost component
- Labels are extended along the arcs using label extension functions to create longer partial paths
- Dominance rules are used to discard non-promising (non Pareto-optimal) labels

Label extension: example

Consider a label $E_i = (Z_i, L_i, F_i, S_i)$ with:

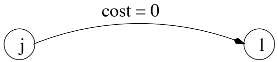
- Z_i : reduced cost of the partial path
- L_i : number of landings in current duty up to i
- F_i : flying time in current duty up to i
- S_i : span of current duty up to i

– Consider a flight arc (i, j) for 90–minute flight f



$E_i = (225, 1, 250, 295)$ $E_j = (-275, 2, 340, 385)$

– Consider a rest arc (j, l) for a 12–hour rest



$E_j = (-275, 2, 340, 385)$ $E_l = (-165, 0, 0, 0)$

Dominance rule: example

- Consider three labels associated with partial paths ending at the same node
  E_1=(-275, 2, 340, 385), E_2=(-320, 2, 225, 360), E_3=(-100, 1, 200, 245)
- Assuming that all extension functions are non-decreasing
  $\Longrightarrow$ E_2 dominates E_1 which can be discarded
  $\Longrightarrow$ E_2 does not dominate E_3 and vice versa
- Labels E_2 and E_3 must be kept
- More complex dominance rule in practice because extension functions are not necessarily non-decreasing

Heuristics

In practice

- When there are many resources, use an aggressive dominance rule (eliminate labels that should be kept)
- To avoid tailing-off, stop column generation prematurely when the objective function decrease in the last iterations is insufficient

Branch-and-price

Column generation is embedded into a branch-and-bound tree to obtain integer solutions $\Longrightarrow$ Branch-and-price method

Branching decisions for exact method

- Not easy and efficient to impose $y_p = 0$
- Impose or forbid two flights to be covered consecutively (inter-task or follow-on branching)

Branching decisions for heuristic method

- Set $y_p = 1$ for the highest fractional variable y_p
- Set the highest fractional inter-task flow to 1

Branch-and-price

Column generation is embedded into a branch-and-bound tree to obtain integer solutions $\Longrightarrow$ Branch-and-price method

Branching decisions for exact method

- Not easy and efficient to impose $y_p = 0$
- Impose or forbid two flights to be covered consecutively (inter-task or follow-on branching)

Branching decisions for heuristic method

- Set $y_p = 1$ for the highest fractional variable y_p
- Set the highest fractional inter-task flow to 1

Branch-and-price

Column generation is embedded into a branch-and-bound tree to obtain integer solutions $\Longrightarrow$ Branch-and-price method

Branching decisions for exact method

- Not easy and efficient to impose $y_p = 0$
- Impose or forbid two flights to be covered consecutively (inter-task or follow-on branching)

Branching decisions for heuristic method

- Set $y_p = 1$ for the highest fractional variable y_p
- Set the highest fractional inter-task flow to 1

Rolling horizon procedure

Saddoune et al. (2009) propose to solve the crew pairing problem directly over the month using a rolling horizon procedure

- Divide the month into overlapping time slices
- In chronological order, solve the problem restricted to each time slice taking into account the solution of the previous slice
- This solution imposes initial conditions for the current slice problem

Computational results

Saddoune, Desaulniers, Soumis (2009). *Aircrew pairings with possible repetitions of the same flight number.* Submitted.

Irregular instance characteristics

Instance	Flights Daily	Flights Weekly	Flights Monthly	Bases	Stations
$I1$	21	175	1011	3	26
$I2$	39	338	1463	3	35
$I3$	50	412	1793	3	41
$I4$	146	1202	5466	3	49
$I5$	158	1229	5639	3	34
$I6$	162	1274	5755	3	52
$I7$	206	1637	7527	3	54

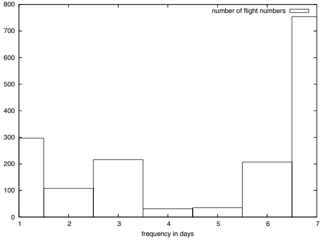


Figure: The weekly regularity

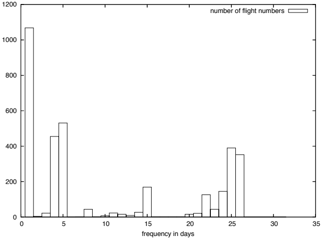


Figure: The monthly regularity

Solution fat (additional credited flying time)

Instance	3P	RH	Reduction
$I1$	9.2%	6.7%	27.2%
$I2$	13.7%	8.5%	40.0%
$I3$	10.9%	7.6%	30.2%
$I4$	7.3%	5.0%	31.5%
$I5$	2.5%	1.2%	52.0%
$I6$	4.2%	3.2%	23.8%
$I7$	4.1%	2.7%	34.1%
Average			34.1%

3P: three-phase RH: rolling horizon

CPU times (in minutes)

Instance	3P Daily	3P Weekly	3P Monthly	3P Total	RH	Reduction
$I1$	< 0.1	0.2	3.4	3.6	3.3	8.3%
$I2$	0.2	1.7	6.8	8.7	5.4	37.9%
$I3$	0.5	3.8	18.2	22.5	15.9	29.3%
$I4$	10.8	200.0	823.5	1034.3	756.6	26.8%
$I5$	2.6	89.1	284.1	375.8	222.9	40.7%
$I6$	3.4	101.7	313.7	418.8	292.7	30.1%
$I7$	4.4	140.2	535.3	679.9	493.0	27.5%
Average						28.4%

Flight number repetitions

Regular instances of weekly problem

	Fat			Pairings with repetitions	
Instance	NR	WR	Reduction	No.	%
$R1$	6.9%	7.5%	−8.7%	36	100.0%
$R2$	20.9%	19.8%	5.3%	26	38.5%
$R3$	26.7%	10.3%	61.4%	34	51.3%
$R4$	5.6%	4.6%	17.9%	51	61.8%
$R5$	4.1%	4.0%	2.4%	51	26.4%
$R6$	4.8%	4.6%	4.2%	63	38.5%
$R7$	7.2%	5.2%	27.8%	77	25.4%
Average	10.9%	8.0%	15.8%	48.3	48.8%

Outline

Crew assignment problem

Separable by

- Crew member category (pilot, copilot, flight director, and flight attendant)
- Aircraft type or family of types
- Crew base

- Each crew member is assigned to a base and must cover pairings assigned to this base
- A schedule is a sequence of pairings, rests and days off that may also contain vacation and training periods
- A schedule is feasible if it satisfies all safety and collective agreement rules, including those pertaining to crew pairings

Crew assignment problem

Separable by

- Crew member category (pilot, copilot, flight director, and flight attendant)
- Aircraft type or family of types
- Crew base

- Each crew member is assigned to a base and must cover pairings assigned to this base
- A schedule is a sequence of pairings, rests and days off that may also contain vacation and training periods
- A schedule is feasible if it satisfies all safety and collective agreement rules, including those pertaining to crew pairings

Examples of rules

- Minimum and maximum number of credited hours per month
- Minimum number of days off per month
- Minimum rest time between two pairings
- Maximum number of consecutive working days
- Valid day off patterns for the month (2 consecutive days at least 4 times, including one weekend)

Three construction modes

Bidline

- Anonymous schedules with equity (credited hours and days off) are built first
- Crew members bid on these schedules
- Schedules are assigned to the members according to their bids and seniority

Rostering

- Takes into account
  - Vacation and training periods
  - Unfinished pairings from the previous month
  - Employees' preferences (in some cases)
- Personalized schedules that maximize the total of the employees' preferences or balance credited hours and days off

Three construction modes

Bidline

- Anonymous schedules with equity (credited hours and days off) are built first
- Crew members bid on these schedules
- Schedules are assigned to the members according to their bids and seniority

Rostering

- Takes into account
  - Vacation and training periods
  - Unfinished pairings from the previous month
  - Employees' preferences (in some cases)
- Personalized schedules that maximize the total of the employees' preferences or balance credited hours and days off

Preferential bidding

- Takes into account
  - Vacation and training periods
  - Unfinished pairings from the previous month
  - Sophisticated weighted preferences
    - 1000 to fly to Paris
    - 200 for a day off on the 22nd
    - 2000 for not working with Guy
- Sequential construction of schedules that maximize the total of the weights obtained by each employee from the most senior to the most junior

Crew assignment problem statement

- Given
  - Set of pairings
  - For each pairing
    - Total number of employees required
    - Minimum number of employees per language (flight attendants)
  - Set of available crew members
  - For each crew member (except for bidline)
    - Vacation and training periods
    - Spoken languages (if needed)
    - Weights for preferences (if needed)
  - Safety and working rules

- Construct feasible schedules according to the selected mode
- Such that
  - A schedule is constructed for each employee
  - Each pairing is covered with the right number of employees (total and minimum per language)

Remark

A typical crew assignment problem instance is defined for a one-month dated flight schedule and solved one month ahead of time

- Construct feasible schedules according to the selected mode
- Such that
  - A schedule is constructed for each employee
  - Each pairing is covered with the right number of employees (total and minimum per language)

Remark

A typical crew assignment problem instance is defined for a one-month dated flight schedule and solved one month ahead of time

Bidline

- Jarrah and Diamond (1997): a priori column generation
- Campbell et al. (1997): simulated annealing
- Christou et al. (1998): genetic algorithm
- Weir and Johnson (2004): MIP three-phase method
- Boubaker et al. (2010): column generation/dynamic constraint aggregation

Rostering

- Set partitioning approaches
  - Ryan (1992), Day and Ryan (1997)
- Column generation methods
  - Gamache and Soumis (1998), Gamache et al. (1999), Kohl and Karisch (2004)

Preferential bidding

- Column generation methods
  - Gamache et al. (1998), Achour et al. (2007)

Rostering

- Set partitioning approaches
  - Ryan (1992), Day and Ryan (1997)
- Column generation methods
  - Gamache and Soumis (1998), Gamache et al. (1999), Kohl and Karisch (2004)

Preferential bidding

- Column generation methods
  - Gamache et al. (1998), Achour et al. (2007)

Bidline for pilots: set partitioning type model

$$\textit{Minimize} \quad \sum_{b \in B} c_b x_b$$

$$s.t. \quad \sum_{b \in B} a_{pb} x_b = 1, \quad \forall p \in P$$

$$\sum_{b \in B} x_b = m$$

$$x_b \in \{0, 1\}, \quad \forall b \in B$$

P: set of pairings to cover

B: set of feasible bidlines

c_b: cost of bidline $b \in B$

a_{pb}: binary parameter taking value 1 if bidline $b \in B$ contains pairing p, and 0 otherwise

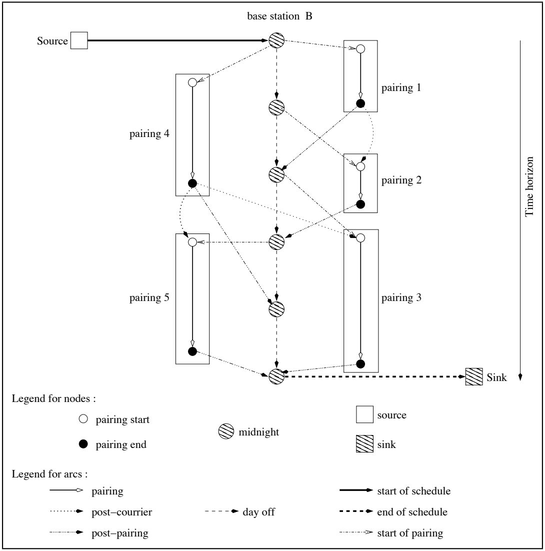
m: number of available pilots

x_b: binary variable equal to 1 if bidline b is selected and 0 otherwise

Column generation for bidline

Subproblems

- Anonymous schedules yield a single subproblem
- In Boubaker et al. (2010), several subproblems because of a complex objective function (minimizing a weighted sum of the variances of the number of days off per schedule and of the number of credited hours per schedule)
- Subproblems are shortest path problems with resource constraints defined on an acyclic network
- Between 4 and 8 resources for a real-life problem


base station B
Source
pairing 1
pairing 2
pairing 3
pairing 4
pairing 5
Sink
Time horizon
Legend for nodes :
pairing start
pairing end
midnight
source
sink
Legend for arcs :
pairing
post–courrier
post–pairing
day off
start of schedule
end of schedule
start of pairing

Dynamic constraint aggregation (DCA)

Introduced in

Elhallaoui, Villeneuve, Soumis, Desaulniers (2005). Dynamic aggregation of set partitioning constraints in column generation, *Operations Research* 53, 632-645.

- Aims at reducing degeneracy in linear programs with set partitioning constraints by working with lower-dimensional bases
- Can be combined with column generation (reduce degeneracy in RMP)

Dynamic constraint aggregation (DCA)

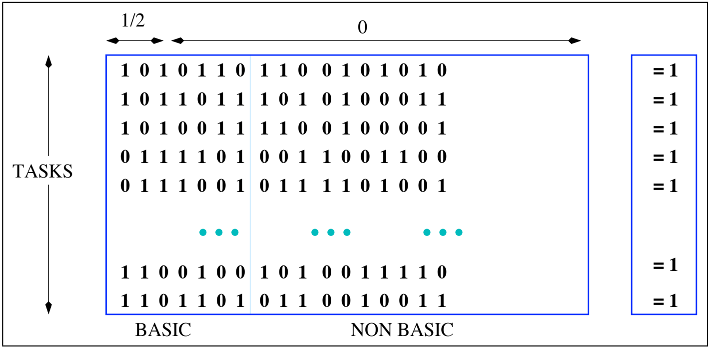
Introduced in

Elhallaoui, Villeneuve, Soumis, Desaulniers (2005). Dynamic aggregation of set partitioning constraints in column generation, *Operations Research* 53, 632-645.

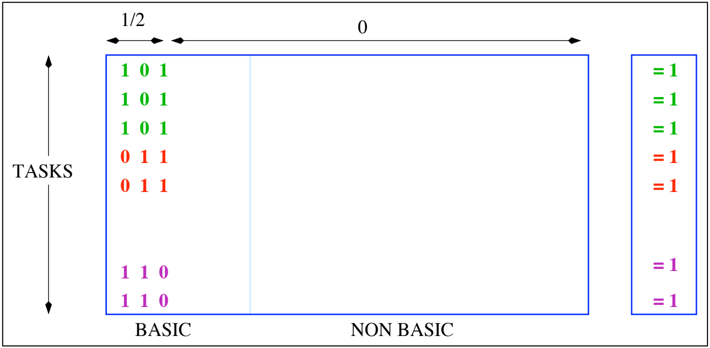
- Aims at reducing degeneracy in linear programs with set partitioning constraints by working with lower-dimensional bases
- Can be combined with column generation (reduce degeneracy in RMP)

Basic concepts of constraint aggregation

- Only for set partitioning constraints
- Tasks are partitioned into clusters $\Longrightarrow$ task partition



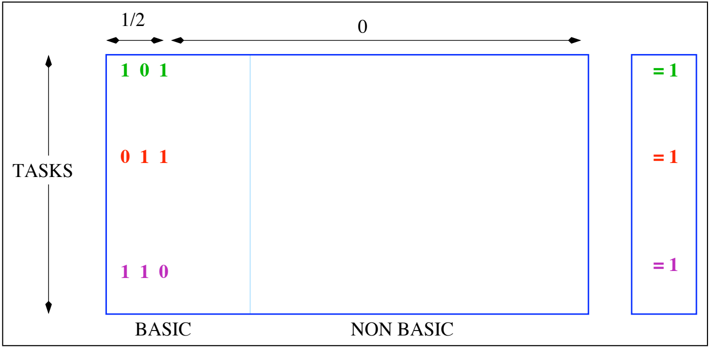
Task clusters



- Task partition built according to a subset of columns
- One task cluster per color (identical rows)
- Higher degeneracy generally yields less clusters

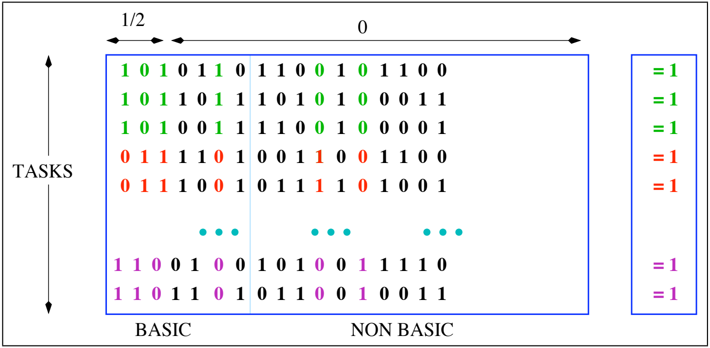
ARMP

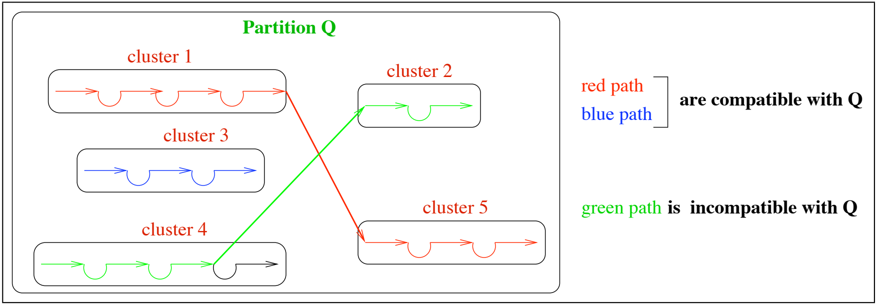
- A cluster is represented by a single constraint in the Aggregated Restricted Master Problem (ARMP)



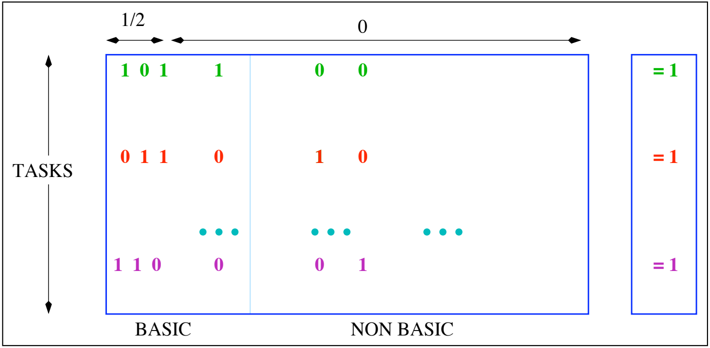
Compatible columns

- A column is compatible with a partition if, for each cluster, it covers all its tasks or none of them




Partition Q
cluster 1
cluster 2
cluster 3
cluster 4
cluster 5
red path
blue path
are compatible with Q
green path is incompatible with Q

- The ARMP contains only compatible columns



Algorithm

– **Set an initial task partition**

– **Construct the ARMP**

Solve the ARMP

→ **Primal solution & Aggregated dual values**

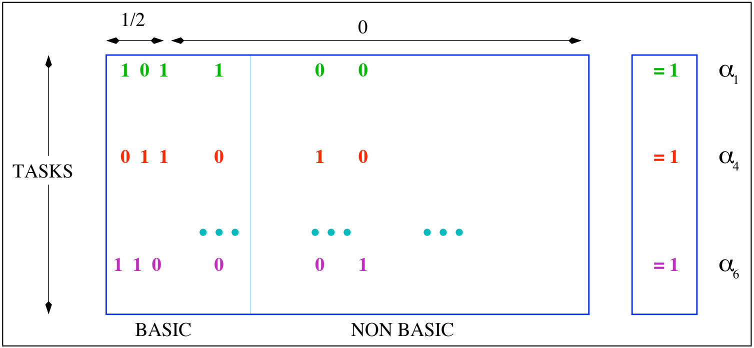
Disaggregate dual values

Solve the subproblems

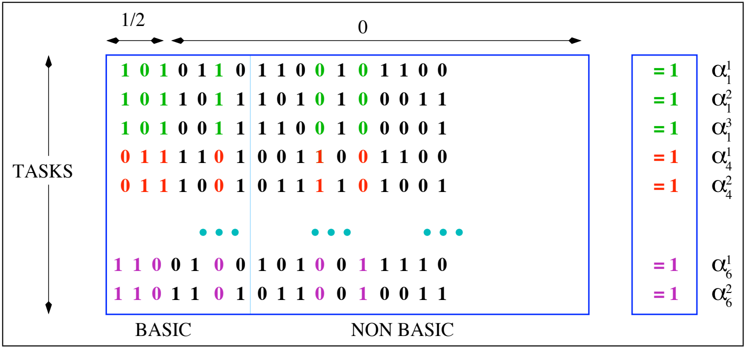
Change the partition ? **No** (return to Solve the ARMP)

– **Update partition (disaggregate or aggregate)** (return to Construct the ARMP)

Aggregated dual values



Disaggregated dual values



Dual variable disaggregation

- L: set of cluster indices ; W_l: set of tasks in cluster l
- P: selected subset of incompatible columns

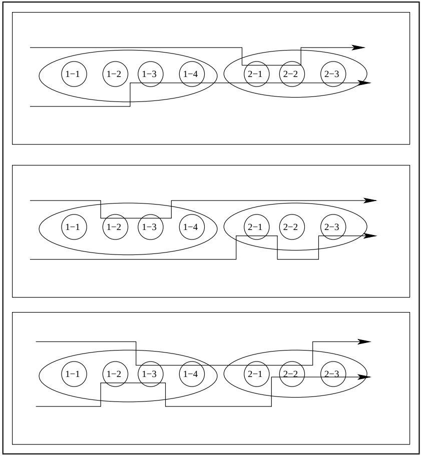
$$\alpha_l = \sum_{w \in W_l} \alpha_l^w, \qquad \forall l \in L \tag{5}$$

$$c_p \geq \sum_{l \in L} \sum_{w \in W_l} a_{lp}^w \alpha_l^w, \qquad \forall p \in P \tag{6}$$

- (5) ensure the same (nonnegative) reduced cost for all compatible variables in the ARMP
- (6) ensure a nonnegative reduced cost for the subset P of incompatible variables

- Dual variables can be disaggregated by solving a linear system of equalities and inequalities
- Difficult to solve in general
- With assumptions on the columns in subset P, solving this system becomes equivalent to solving a series of shortest path problems

Incompatibility types



(Arbitrary) ordered tasks in clusters

p_1 : S-incompatible

p_2 : E-incompatible

p_3 : M-incompatible

p_4 : SE-incompatible

p_5 : ES-incompatible

p_6 : O-incompatible

Variable substitution

$$\beta_l^w = \sum_{j=1}^{w} \alpha_l^j, \quad \forall w \in W_l,\ l \in L$$

For example, for a cluster with 4 ordered tasks:

$$\beta_1^1 = \alpha_1^1 \qquad \beta_1^2 = \alpha_1^1 + \alpha_1^2$$

$$\beta_1^3 = \alpha_1^1 + \alpha_1^2 + \alpha_1^3 \qquad \beta_1^4 = \alpha_1^1 + \alpha_1^2 + \alpha_1^3 + \alpha_1^4$$

Constraints (5) become

$$\beta_l^{|W_l|} = \alpha_l, \ \forall l \in L$$

Variable substitution

$$\beta_l^w = \sum_{j=1}^{w} \alpha_l^j, \quad \forall w \in W_l,\ l \in L$$

For example, for a cluster with 4 ordered tasks:

$$\beta_1^1 = \alpha_1^1 \qquad \beta_1^2 = \alpha_1^1 + \alpha_1^2$$

$$\beta_1^3 = \alpha_1^1 + \alpha_1^2 + \alpha_1^3 \qquad \beta_1^4 = \alpha_1^1 + \alpha_1^2 + \alpha_1^3 + \alpha_1^4$$

Constraints (5) become

$$\beta_l^{|W_l|} = \alpha_l, \ \forall l \in L$$

Transformation of constraints (6)

Category	Transformed constraint		
p is S-incompatible	β_i^m	$\leq$	$c_p - \sum_{l \in L_p} \alpha_l$
p is E-incompatible	$-\beta_i^{m-1}$	$\leq$	$c_p - \alpha_i - \sum_{l \in L_p} \alpha_l$
p is M-incompatible	$\beta_i^n - \beta_i^{m-1}$	$\leq$	$c_p - \sum_{l \in L_p} \alpha_l$
p is SE-incompatible	$\beta_i^m - \beta_i^{m+n-1}$	$\leq$	$c_p - \alpha_i - \sum_{l \in L_p} \alpha_l$
p is ES-incompatible	$\beta_j^n - \beta_i^{m-1}$	$\leq$	$c_p - \alpha_i - \sum_{l \in L_p} \alpha_l$

- Limiting the subset P to these types, the linear system (5)–(6) transforms into a set of inequalities corresponding to the constraints of the dual of a shortest path problem
- This dual might be infeasible $\Longrightarrow$ Reduce P

Transformation of constraints (6)

Category	Transformed constraint		
p is S-incompatible	β_i^m	$\leq$	$c_p - \sum_{l \in L_p} \alpha_l$
p is E-incompatible	$-\beta_i^{m-1}$	$\leq$	$c_p - \alpha_i - \sum_{l \in L_p} \alpha_l$
p is M-incompatible	$\beta_i^n - \beta_i^{m-1}$	$\leq$	$c_p - \sum_{l \in L_p} \alpha_l$
p is SE-incompatible	$\beta_i^m - \beta_i^{m+n-1}$	$\leq$	$c_p - \alpha_i - \sum_{l \in L_p} \alpha_l$
p is ES-incompatible	$\beta_j^n - \beta_i^{m-1}$	$\leq$	$c_p - \alpha_i - \sum_{l \in L_p} \alpha_l$

- Limiting the subset P to these types, the linear system (5)–(6) transforms into a set of inequalities corresponding to the constraints of the dual of a shortest path problem
- This dual might be infeasible $\Longrightarrow$ Reduce P

Algorithm (once again)

– Set an initial task partition

– Construct the ARMP

Solve the ARMP

→ **Primal solution & Aggregated dual values**

Disaggregate dual values

Solve the subproblems

Change the partition ? **No** (→ back to Solve the ARMP)

– Update partition (disaggregate or aggregate) (→ back to Construct the ARMP)

Partition update

- A task partition is built using a subset C of columns
- For the initial partition, C is composed of columns (feasible or not) obtained from
  - A heuristic solution
  - Logical reasoning
- Disaggregate the partition when there exist incompatible variables with large negative reduced costs (compared to the reduced costs of the compatible variables)
- Aggregate the partition when degeneracy becomes important

- When disaggregating
  - The current subset C is augmented by a small number of incompatible columns that have negative reduced costs
- When aggregating
  - The current subset C is replaced by the set of variables with a positive value in the current master problem solution

Multi-phase DCA: Concepts

Introduced in

Elhallaoui, Metrane, Soumis, Desaulniers (2010). Multi-phase dynamic aggregation for set partitioning type problems, *Mathematical Programming A* 123(2), 345-370.

- In practice, incompatible variables often price out favorably
  - May yield fast disaggregation
- Partial pricing strategy that favors slow disaggregation
- With each column, associate a number of incompatibilities
  - Approximation of the number of additional clusters needed to become compatible
- In phase k: price only variables with k incompatibilities or less

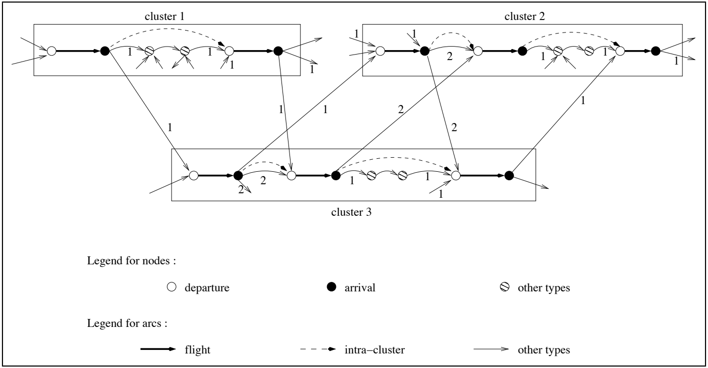
Multi-phase DCA: Concepts

Introduced in

Elhallaoui, Metrane, Soumis, Desaulniers (2010). Multi-phase dynamic aggregation for set partitioning type problems, *Mathematical Programming A* 123(2), 345-370.

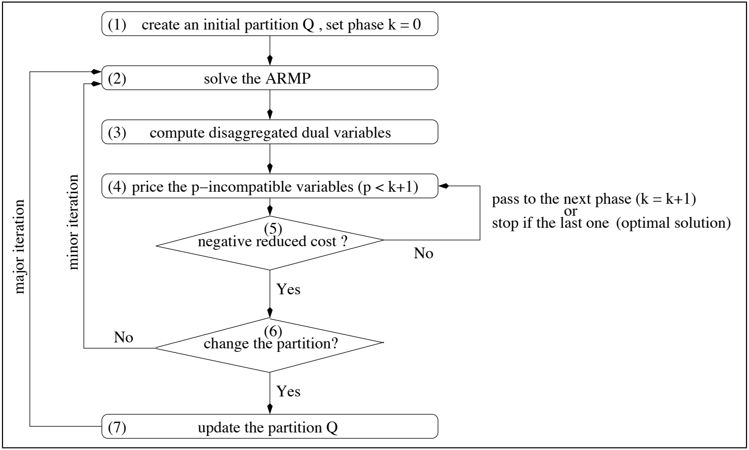
- In practice, incompatible variables often price out favorably
  - May yield fast disaggregation
- Partial pricing strategy that favors slow disaggregation
- With each column, associate a number of incompatibilities
  - Approximation of the number of additional clusters needed to become compatible
- In phase k: price only variables with k incompatibilities or less

Number of incompatibilities



A resource is used to limit the number of incompatibilities in a path

MPDCA algorithm



Computational results

Boubaker, Desaulniers, Elhallaoui (2010). Bidline scheduling with equity by heuristic dynamic constraint aggregation. *Transportation Research Part B: Methodological*, 44, 50–61.

- Initial task partition computed from a tabu search solution
- Heuristic branching (column fixing)

Solution process statistics

	Times (s)			Numbers of		
Instance	Total	RMP	SP	Iter.	Nodes	Fract. var.
BP heuristic						
1187/228	3852	2503	1341	2607	228	1188
1507/289	9230	6715	2500	3260	287	1488
2165/416	43 625	36 174	7415	5131	416	2154
2924/564	95 215	86 228	8927	6408	563	2914
DCA heuristic						
1187/228	348	6	237	861	156	344
1507/289	480	7	341	1157	191	374
2165/416	1279	35	992	1896	317	617
2924/564	3076	61	2345	3149	440	608

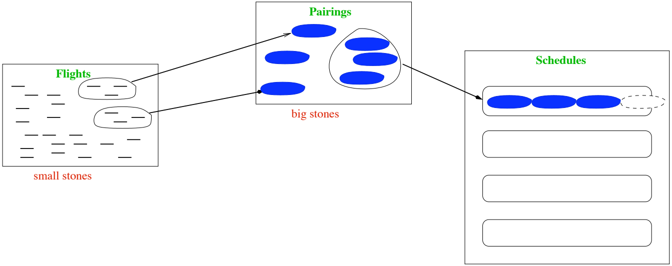
Solution quality statistics

Instance	Credited hours				Days off			
	Mean	Min	Max	Var.	Mean	Min	Max	Var.
BP heuristic								
1187/228	75.2	71.1	80.0	4.12	13.7	11	16	1.59
1507/289	75.2	71.2	80.7	3.50	13.7	11	16	1.32
2165/416	75.2	71.0	82.7	3.17	13.7	11	17	1.25
2924/564	75.1	71.0	81.1	6.50	13.8	11	17	2.04
DCA heuristic								
1187/228	75.2	73.0	79.0	1.17	13.5	12	15	0.59
1507/289	75.2	72.6	78.8	1.44	13.4	12	16	0.69
2165/416	75.2	73.0	78.8	1.61	13.4	12	16	0.69
2924/564	75.1	71.9	79.0	1.64	13.5	12	16	0.77

Average value of fixed variables: 0.91 with DCA, 0.71 with BP

Outline

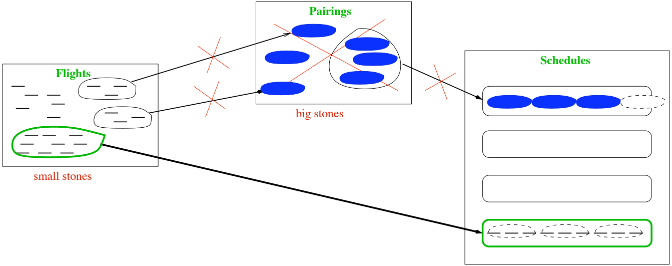
Traditional sequential approach



Weaknesses

- Pairings are built without fully taking into account schedule feasibility rules
- Less choices to construct schedules

Integrated approach



Strengths

- Schedules are built directly from the flights (more choices)
- Schedule feasibility rules are considered throughout the process
- Crew assignment is not subject to a separation by crew base

Literature review

Only **one** paper on integrated crew scheduling

- Zeghal and Minoux (2006)
  - Exact and heuristic branch-and-bound methods
  - Assume that
    - All duties can be enumerated a priori
    - Deadheads can be introduced as needed without additional costs
    - Small-sized instances (up to 210 flights or 40 crew members)

Model for integrated problem (bidline, pilots)

$$\text{Minimize} \quad \sum_{b \in B} \sum_{s \in S^b} c_s x_s + \beta \sum_{b \in B} y_b$$

$$\text{s.t.} \quad \sum_{b \in B} \sum_{s \in S^b} a_{fs} x_s = 1, \quad \forall f \in F$$

$$\sum_{s \in S^b} x_s - y_b \leq q_b, \quad \forall b \in B$$

$$x_s \in \{0,1\}, \quad \forall b \in B, \; s \in S^b$$

$$y_b \geq 0, \quad \forall b \in B$$

F: set of flights to cover;

B: set of crew bases;

q_b: number of pilots available at base b;

β: penalty cost for each additional pilot;

S^b: set of feasible schedules for pilots at base b;

c_s: cost of schedule s;

a_{fs}: binary parameter equal to 1 if flight f is actively covered in schedule s and 0 otherwise;

x_s: binary variable taking value 1 if schedule s is selected and 0 otherwise;

y_b: surplus variable indicating the number of extra pilots required at the base b.

Integrated problem: Complexity?

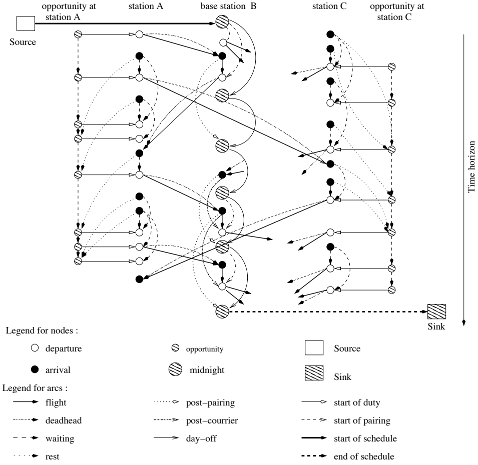
Difficulties

1. Large number of set partitioning constraints
2. Large number of tasks per path yielding high degeneracy

Size	Pairing (weekly)	Assignment	Integrated
200 flights/day	1400	600	6000
1000 flights/day	7000	3000	30000

Problem	Tasks	Paths	Tasks/path
Pairing (weekly)	flights	pairings	$\sim$10
Assignment	pairings	schedules	$\sim$5
Integrated	flights	schedules	$\sim$50

Column generation: subproblems



- One-month horizon
- One subproblem per crew base
- Between 9 and 18 resources

Bi-dynamic constraint aggregation

Introduced in

Elhallaoui, Desaulniers, Metrane, Soumis (2008). Bi-dynamic constraint aggregation and subproblem reduction, *Computers & Operations Research* 35(5), 1713-1724.

- With MPDCA, most of the computational time is spent solving the subproblems
- To avoid fast disaggregation, forbid the pricing of columns that would force the disaggregation of certain clusters
- This is another partial pricing strategy

Bi-dynamic constraint aggregation

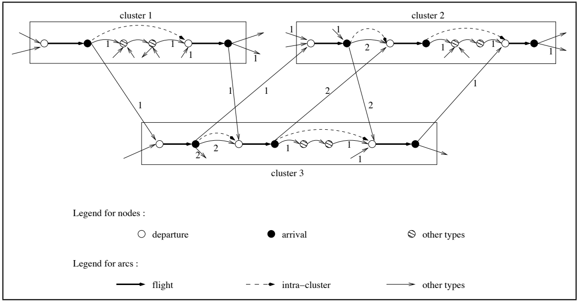
Introduced in

Elhallaoui, Desaulniers, Metrane, Soumis (2008). Bi-dynamic constraint aggregation and subproblem reduction, *Computers & Operations Research* 35(5), 1713-1724.

- With MPDCA, most of the computational time is spent solving the subproblems
- To avoid fast disaggregation, forbid the pricing of columns that would force the disaggregation of certain clusters
- This is another partial pricing strategy

Main ideas

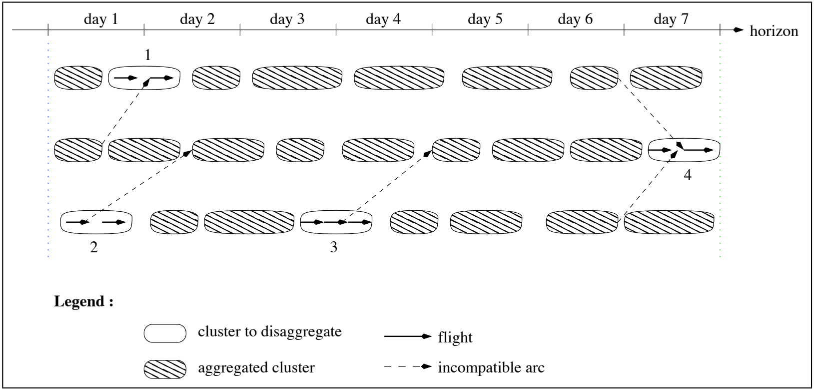
- Reduce the subpoblem networks according to current task partition
  - Select a certain number of task clusters
  - Remove all incompatible arcs associated with these clusters
- If no negative reduced cost columns, solve with complete networks



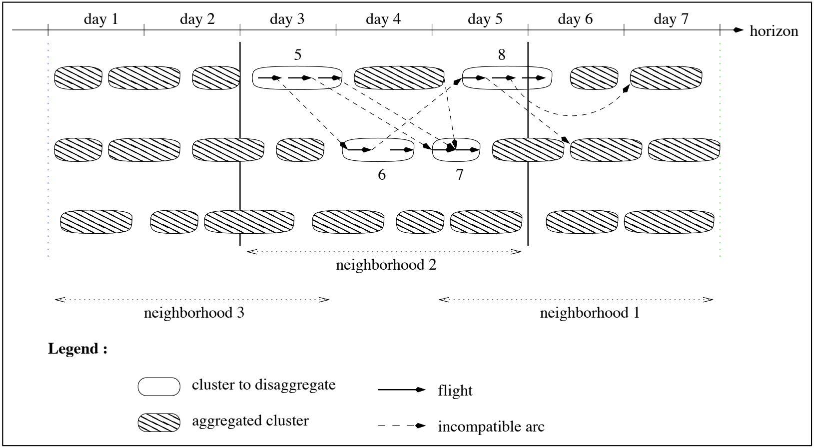
Cluster selection based on

- Aggregated dual values of the clusters
- Cluster reduced costs
- Neighborhoods to increase interaction between generated columns

Cluster selection without neighborhoods



Cluster selection based on neighborhoods



Algorithm 1 : BDCA with neighborhoods (BDCA-N)

Select an initial neighborhood N
Set $k := 0$, $Z_{old} := \infty$, $Z := \infty$
Create an initial partition Q and build the ARMP
Solve the ARMP with artificial variables
repeat
    Select a subset of clusters U according to RL and N
    $negRedCostCols := LocalSearch(Z, Q, k, U)$
    if $negRedCostCols = false$ **then**
        $k := k + 1$
    else if $Z_{old} - Z \leq \delta$ **then**
        Select a new neighborhood N
    $Z_{old} := Z$
until $k > k_{max}$

Algorithm 2 :*LocalSearch*(Z, Q, k, U)

1: *curIter* $:= 1$
2: **repeat**
3:    Compute disaggregated dual values
4:    *redNetwork* := *true*
5:    Reduce the subproblems according to U and solve them in phase k
6:    **if** *negative reduced cost columns found* **then**
7:       Update partition Q (if needed) and the ARMP
8:       Solve the ARMP and update Z
9:       *curIter* := *curIter* $+ 1$
10:    **else if** *redNetwork* = *true* and $k > 0$ **then**
11:       *redNetwork* := *false*
12:       Solve the complete subproblems in phase k and go to Step 6
13:    **else**
14:       **return** *false*
15: **until** $curIter > maxIter_k$
16: **return** *true*

Computational results

Saddoune, Desaulniers, Soumis (2010). *Integrated airline crew scheduling: A bi-dynamic constraint aggregation method using neighborhoods.* Submitted.

- Initial task partition from the crew pairing solution
- Phases 0 and 1 only
- Column generation stops if objective value decreases by less than 0.01% in 10 iterations
- BDCA-N-1 version forces phase 1 if premature halt in phase 0

Computational results

Saddoune, Desaulniers, Soumis (2010). *Integrated airline crew scheduling: A bi-dynamic constraint aggregation method using neighborhoods.* Submitted.

- Initial task partition from the crew pairing solution
- Phases 0 and 1 only
- Column generation stops if objective value decreases by less than 0.01% in 10 iterations
- BDCA-N-1 version forces phase 1 if premature halt in phase 0

Instance characteristics

	Number of		
Instance	flights	bases	stations
$I1$	1011	3	26
$I2$	1463	3	35
$I3$	1793	3	41
$I4$	5466	3	49
$I5$	5639	3	34
$I6$	5755	3	52
$I7$	7527	3	54

Remark

- For instance $I1$, standard column generation took more than 12 hours of CPU time (sequential approach took 4 minutes)
- For all other instances, computation was stopped after 2 days

Instance characteristics

Instance	Number of flights	Number of bases	Number of stations
$I1$	1011	3	26
$I2$	1463	3	35
$I3$	1793	3	41
$I4$	5466	3	49
$I5$	5639	3	34
$I6$	5755	3	52
$I7$	7527	3	54

Remark

- For instance $I1$, standard column generation took more than 12 hours of CPU time (sequential approach took 4 minutes)
- For all other instances, computation was stopped after 2 days

Comparative results

SEQ: sequential approach (crew pairing, then crew assignment)

	BDCA-N approach			BDCA-N-1 approach		
	CPU ratio	Savings (%)		CPU ratio	Savings (%)	
Instance	BDCA-N/SEQ	Cost	Sched	BDCA-N-1/SEQ	Cost	Sched
$I1$	1.7	5.74	6.06	1.9	5.72	6.06
$I2$	2.5	3.60	8.82	3.2	3.28	5.88
$I3$	3.0	3.07	8.51	5.3	4.46	10.63
$I4$	1.8	3.42	5.51	2.1	4.02	5.51
$I5$	6.0	4.09	2.42	6.6	4.97	3.23
$I6$	3.0	6.75	6.27	4.2	8.78	8.07
$I7$	3.0	1.50	0.98	3.2	2.14	1.63
Average	3.0	4.02	5.51	3.8	4.76	5.85

Average results comparison

	CPU ratio	Savings (%)	
Instance	BDCA-N/SEQ	Cost	Sched
MPDCA	6.8	3.37	5.54
BDCA	2.6	2.63	4.63
BDCA-N	3.0	4.02	5.51
BDCA-N-1	3.8	4.76	5.85

Outline

Conclusion

- For several decades, airline crew scheduling has been a fertile ground for operations research
- Operations research tools have helped reducing crew costs and increasing crew member quality of life
- Research on this topic remains very active because there still remain many challenges

Future work

- Improve dynamic constraint aggregation
  - Dual variable disaggregation
  - Neighborhood exploration
- Experiment IPS (improved primal simplex) method on certain crew scheduling problems
  - IPS is a generalization of DCA to generic linear programs
  - It uses a reduced problem and a complementary problem to identify good incompatible variables
- Integrated crew scheduling in a rostering context (personalized schedules)
  - Pilots only
  - Pilots and copilots simultaneously

Thank you!