Dynamic constraint aggregation

Guy Desaulniers

École Polytechnique de Montréal and GERAD, Canada

School on Column Generation 2014
Paris, France, March 10–14, 2014

Collaborators

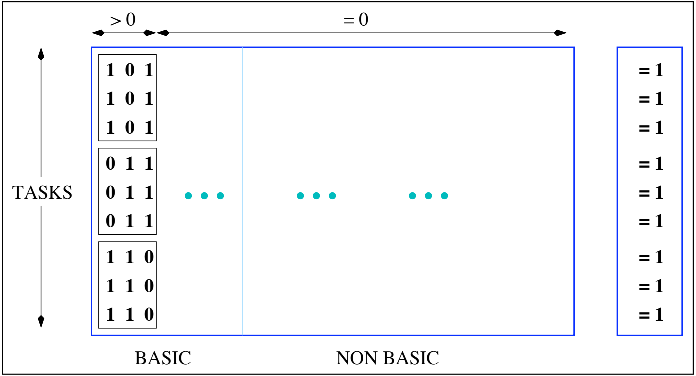
- Khaled Boubaker, MSc student
- Issmail Elhallaoui, PhD student and post-doc
- Alain Hertz, professor
- Abdelmoutalib Metrane, post-doc
- Mohammed Saddoune, PhD student
- Personnel from Ad Opt/Kronos

Outline

Outline

Massive degeneracy

- Several vehicle routing and crew scheduling problems involve tasks that must be accomplished once
- These are modeled using set partitioning constraints
- In a branch-and-price algorithm, these constraints are handled in the restricted master problem (RMP)
- They often yield high degeneracy that slows down the solution process of the RMP at each iteration



When the average number of tasks per path is high (say, > 10) and there are many constraints (e.g., more than 1000), degeneracy substantially slows down the solution process

Simultaneous vehicle and crew scheduling problem in urban mass transit systems (VCSP)

- Single depot, homogeneous fleet
- Each trip must be covered by exactly one bus
- Each segment assigned to exactly one driver
- Work rules satisfied (breaks, duty length, ...)
- Minimize total costs including bus and driver operational and fixed costs

Haase, Desaulniers, Desrosiers (2001). Simultaneous vehicle and crew scheduling in urban mass transit systems, *Transportation Science* 35, 286-303.

Linear relaxation of VCSPs – Standard column generation

Tasks	400	800	1200	1600
MP constraints	415	833	1252	1662
MP CPU time (s)	42	884	5835	19711
Total CPU time (s)	52	993	6402	21508

- Number of tasks covered per column = 20 to 25
- When the number of tasks is multiplied by 4, total CPU time is multiplied by 400 !

Intuition

- A trip is divided into consecutive segments
- All these segments are assigned to the same bus
- A driver cannot often change buses
- High chances that the driver on the first segment of a trip will also drive the other segments

- Define a single set partitioning constraint for the segments in a trip
- Force paths to cover all segments in a trip
- Revise segment clustering if needed

Outline

Dynamic constraint aggregation (DCA)

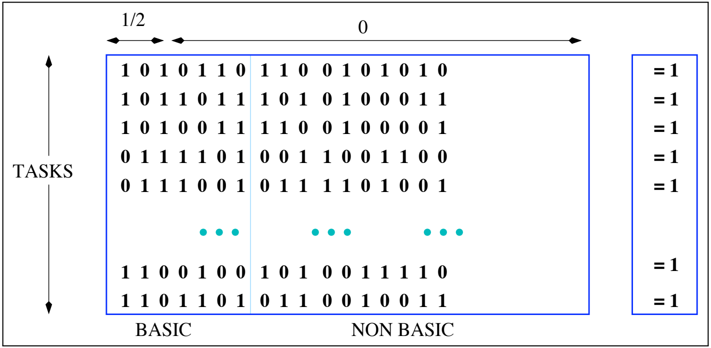
Introduced in

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

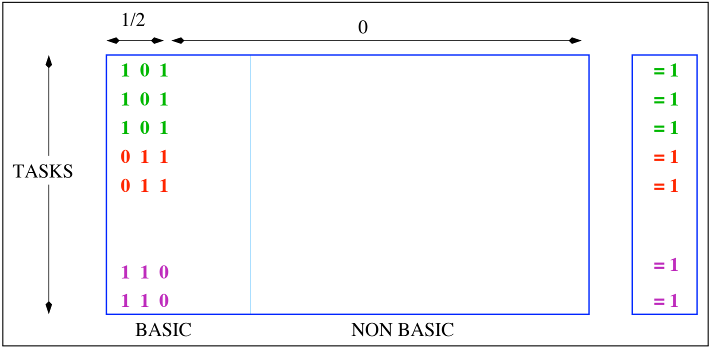
- Aims at reducing degeneracy in linear programs with set partitioning constraints by working with lower-dimensional bases
- Can be combined with column generation (reduce degeneracy in RMP)

Basic concepts of constraint aggregation

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



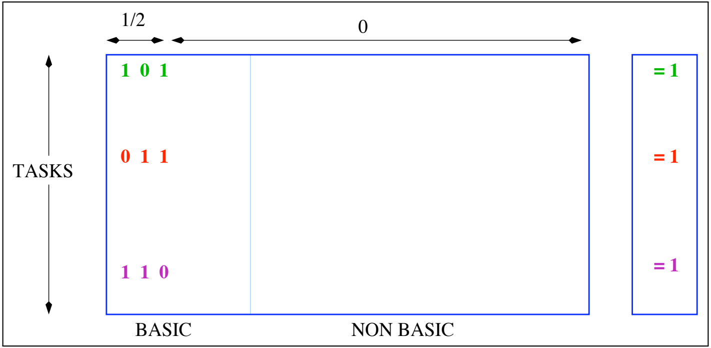
Task clusters



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

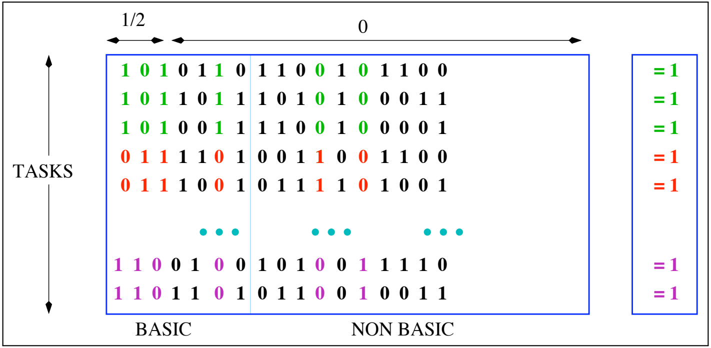
ARMP

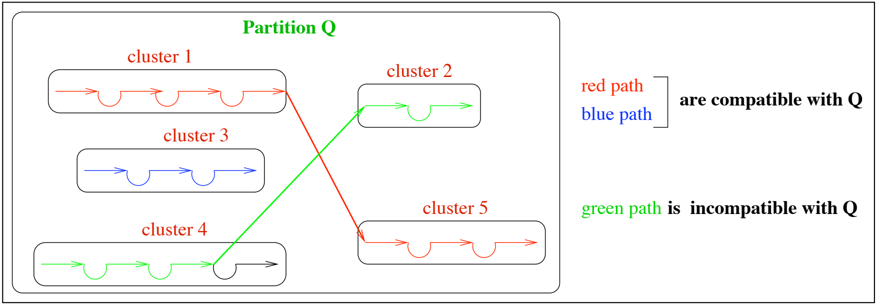
- A cluster is represented by a single constraint in the Aggregated Restricted Master Problem (ARMP)



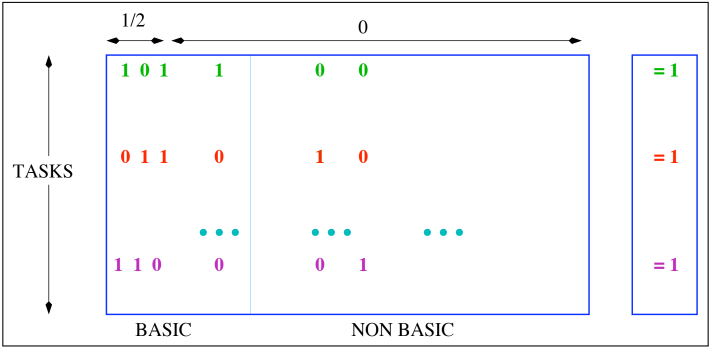
Compatible columns

- 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**

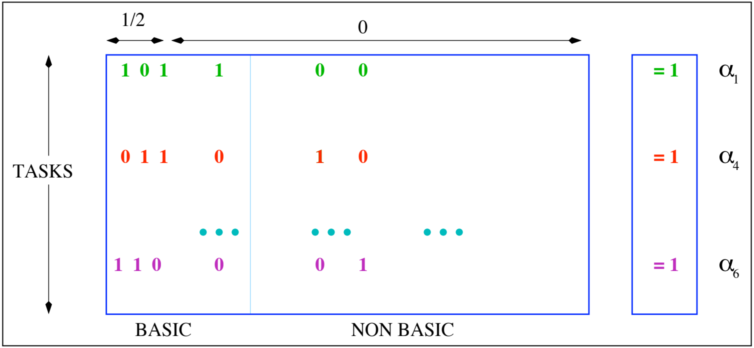
Disaggregate dual values

Solve the subproblems

Change the partition ? **No**

– **Update partition (disaggregate or aggregate)**

Aggregated dual values



Disaggregated dual values

1/2 0

TASKS	1	0	1	0	1	1	0	1	1	0	0	1	0	1	1	0	0	= 1	α_1^1
	1	0	1	1	0	1	1	1	0	1	0	1	0	0	0	1	1	= 1	α_1^2
	1	0	1	0	0	1	1	1	1	0	0	1	0	0	0	0	1	= 1	α_1^3
	0	1	1	1	1	0	1	0	0	1	1	0	0	1	1	0	0	= 1	α_4^1
	0	1	1	1	0	0	1	0	1	1	1	1	0	1	0	0	1	= 1	α_4^2
				• • •						• • •				• • •					
	1	1	0	0	1	0	0	1	0	1	0	0	1	1	1	1	0	= 1	α_6^1
	1	1	0	1	1	0	1	0	1	1	0	0	1	0	0	1	1	= 1	α_6^2

BASIC NON BASIC

Dual variable disaggregation

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

$$\alpha_l = \sum_{w \in W_l} \alpha_l^w, \qquad \forall l \in L \tag{1}$$

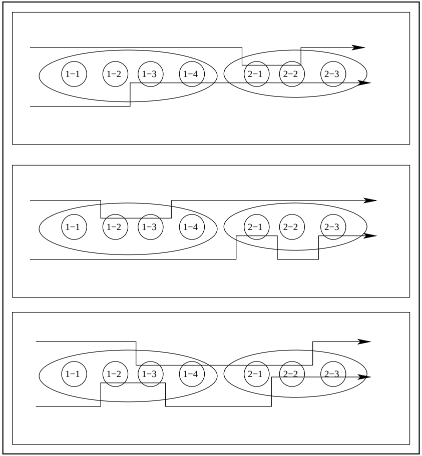
$$c_p \geq \sum_{l \in L} \sum_{w \in W_l} a_{lp}^w \alpha_l^w, \qquad \forall p \in P \tag{2}$$

- (1) ensure the same (nonnegative) reduced cost for all compatible variables in the ARMP
- (2) ensure a nonnegative reduced cost for the incompatible variables in subset P

- Dual variables can be disaggregated by solving a linear system of equalities and inequalities (1)–(2)
- 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 (1) become

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

Transformation of constraints (2)

Category	Transformed constraint
p is S-incompatible	$\beta_i^m \leq c_p - \sum_{l \in L_p} \alpha_l$
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 (1)–(2) 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**

– **Update partition (disaggregate or aggregate)**

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 RMP solution

Outline

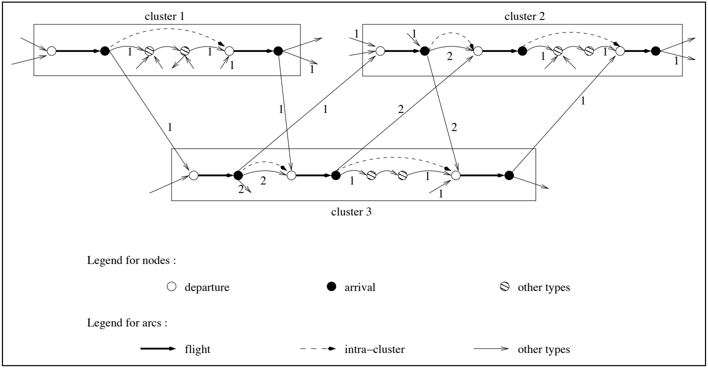
Multi-phase DCA

Introduced in

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

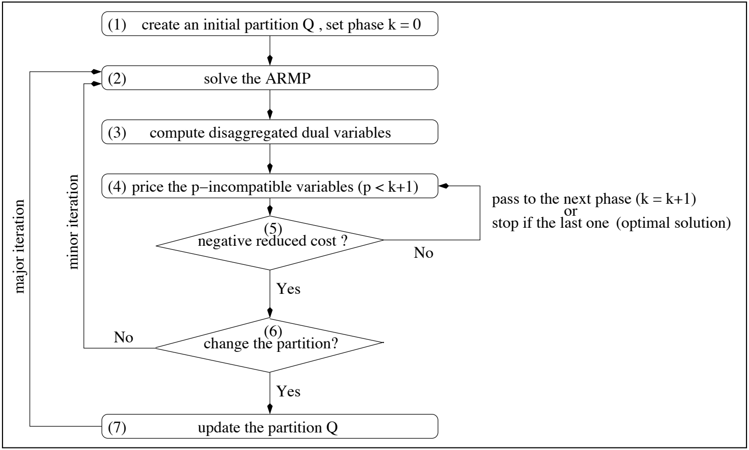
- In practice, incompatible variables often price out favorably
  - May yield fast disaggregation
- Partial pricing strategy that favors slow disaggregation
- With each column, associate a number of incompatibilities
  - Approximation of the number of additional clusters needed to become compatible
-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



Bi-dynamic constraint aggregation (BDCA)

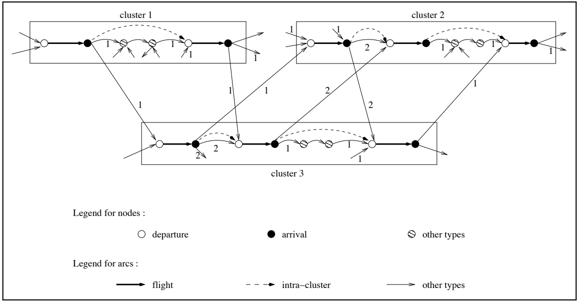
Introduced in

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 (based on dual values)
  - Remove all incompatible arcs associated with these clusters
- If no negative reduced cost columns, solve with complete networks



Outline

Linear relaxation results for VCSP

Trips	120				160				200			
Tasks	1200				1600				2000			
Results	STD	DCA	MPDCA	BDCA	STD	DCA	MPDCA	BDCA	STD	DCA	MPDCA	BDCA
CG iterations	480	906	226	409	618	939	433	765	798	1040	242	720
Avg MP const	1250	763	423	263	1664	1083	549	433	2084	1139	507	445
Part chg	–	295	165	150	–	324	342	199	–	577	226	249
SP time (s)	676	1638	515	108	2393	3980	2688	452	3932	5294	2056	741
MP time (s)	5395	1120	89	60	21903	7152	1021	288	53957	5479	308	404
Total time (s)	6093	2827	647	205	24337	11250	3830	875	57948	11201	2475	1348

STD: Standard column generation
Averages over three instances

From Elhallaoui et al. (2008, 2010)

Integer solution results for VCSP

Trips	80		100		120		160	
Tasks	800		1000		1200		1600	
Results	STD	BDCA	STD	BDCA	STD	BDCA	STD	BDCA
Frac var					269	117	343	111
BB nodes	17	4.6	>38	4.3	>41	6.3	>1	12
LP time (s)	1126	50	4085	298	6093	205	24337	875
Total time (s)	2807	54	>21600	305	>21600	216	>21600	1542

STD: Standard column generation
Averages over three instances

From Elhallaoui et al. (2008)

Bidline scheduling for air pilots

Given a set of pairings (sequences of flights), find anonymous monthly schedules such that

- All pairings are covered by one pilot
- Every pilot is assigned to a feasible schedule
- Security and labor rules are met
- Two-fold objective
  - Minimize standard deviation of number of credited hours per schedule
  - Minimize standard deviation of number of days off per schedule

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

Two solution methods

1. Branch-and-price (BP) heuristic
   - Early termination of column generation
   - Column fixing to derive integer solutions
   - Five subproblems to approximate the objective function
   - Five resources to handle working rules
2. MPDCA heuristic
   - Same setting as above
   - Initial task partition derived from a tabu search heuristic solution

Solution process statistics

Instance	Times (s) Total	RMP	SP	Numbers of 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

From Boubaker et al. (2010)

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

From Boubaker et al. (2010)