The Balance Constraint Family

Christian Bessiere, Emmanuel Hebrard, George Katsirelos, Zeynep Kiziltan, Emilie Picard-Cantin, Claude-Guy Quimper, Toby Walsh

Abstract

The ***Balance*** constraint introduced by Beldiceanu ensures solutions are balanced. This is useful when, for example, there is a requirement for solutions to be fair. ***Balance*** bounds the difference $\boldsymbol{B}$ between the minimum and maximum number of occurrences of the values assigned to the variables. We show that achieving domain consistency on ***Balance*** is NP-hard. We therefore introduce a variant, ***AllBalance*** with a similar semantics that is only polynomial to propagate. We consider various forms of ***AllBalance*** and focus on ***AtMostAllBalance*** which achieves what is usually the main goal, namely constraining the upper bound on $\boldsymbol{B}$. We provide a specialized propagation algorithm, and a powerful decomposition both of which run in low polynomial time. Experimental results demonstrate the promise of these new filtering methods.

Basic notions

- **Number of occurrences ($\mathrm{occ}(v)$) :** Number of times that a value is allocated.
- **Example :**
  Variable $X_{w,d}$: task done by the worker $\boldsymbol{w}$ on day $\boldsymbol{d}$

  Possible values : 

  Domain : worker qualifications
- **Satisfaction of constraint C :** An assignment satisfies C if the set of values allocated obey the relation defined by C.
- **Domain consistency :** When all the values in the domains contribute to a potential solution (solution that satisfies the constraints).
- **Constraint decomposition :** Breakdown of a constraint into multiple simpler constraints.

The Balance Family

- $\mathit{Balance}([X_1, \ldots, X_n], B)$
  $\iff B = \max_{v \in \{X_1,\ldots,X_n\}} \mathrm{occ}(v) - \min_{v \in \{X_1,\ldots,X_n\}} \mathrm{occ}(v)$
- $\mathit{AllBalance}(\mathcal{V}, [X_1, \ldots, X_n], B)$
  $\iff B = \max_{v \in \mathcal{V}} \mathrm{occ}(v) - \min_{v \in \mathcal{V}} \mathrm{occ}(v)$
- $\mathit{AtMostBalance}([X_1, \ldots, X_n], B)$
  $\iff B \geq \max_{v \in \{X_1,\ldots,X_n\}} \mathrm{occ}(v) - \min_{v \in \{X_1,\ldots,X_n\}} \mathrm{occ}(v)$
- $\mathit{AtMostAllBalance}(\mathcal{V}, [X_1, \ldots, X_n], B)$
  $\iff B \geq \max_{v \in \mathcal{V}} \mathrm{occ}(v) - \min_{v \in \mathcal{V}} \mathrm{occ}(v)$
- ***AtLeastBalance*** and ***AtLeastAllBalance***

Examples

	L	M	MR	J	V	S	D
Robert							

$$\Rightarrow \begin{cases} \mathrm{occ}(\ ) = 0, \mathrm{occ}(\ ) = 5 \\ \mathrm{occ}(\ ) = 2 \end{cases}$$

Balance	$B = \max - \min = 5 - 2 \Rightarrow B = 3$
AllBalance	$B = \max - \min = 5 - 0 \Rightarrow B = 5$
AtMostBalance	$B \geq \max - \min = 5 - 2 \Rightarrow B \geq 3$
AtMostAllBalance	$B \geq \max - \min = 5 - 0 \Rightarrow B \geq 5$
AtLeastBalance	$B \leq \max - \min = 5 - 2 \Rightarrow B \leq 3$
AtLeastAllBalance	$B \leq \max - \min = 5 - 0 \Rightarrow B \leq 5$

Results

	Original	"AI"
Balance	NP-hard	Polynomial
AtMostBalance	NP-hard	Polynomial
AtLeastBalance	Polynomial	Polynomial

Filtering algorithm for *AtMostAllBalance*

1. Find a potential solution that satisfies AtMostAllBalance and such that the balance variable $\boldsymbol{B}$ is minimal.
2. Let $q = \min_{v \in \mathcal{V}} \mathrm{occ}(v)$.
3. Run filter algorithm for $\mathrm{GCC}([\mathrm{D}(X_1), \ldots, \mathrm{D}(X_n)], [O_1, \ldots, O_m])$ where $O_i \in [q, q + \max(B)] \quad \forall i$ and mark values that have a support.
4. If no values were filtered, mark all values since they all contribute to a potential solution for AtMostAllBalance.
5. If a value was filtered, as in step 3, mark values with
   - $O_i \in [q+1, q+\max(B)+1] \quad \forall i$ if $\exists$ a Hall set
   - $O_i \in [q-1, q+\max(B)-1] \quad \forall i$ if $\exists$ an unstable set
6. Remove all values that are not marked.

Decompositions of the *AllBalance* family

Decomp.

$\mathrm{GCC}([X_1, \ldots, X_n], [O_1, \ldots, O_m])$
$P = \max(\{O_1, \ldots, O_m\})$
$Q = \min(\{O_1, \ldots, O_m\})$
$B = P - Q$

Implied

$$\text{Decomp.} \quad + \quad \begin{cases} mP - (m-1)B \leq n \\ mQ + (m-1)B \geq n \end{cases}$$

Implied+

$$\text{Implied} \quad + \quad \begin{cases} \sum_{j=1}^{m} \max(P - B, O_j) \leq n \leq \sum_{j=1}^{m} \min(P, O_j) \\ \sum_{j=1}^{m} \min(Q + B, O_j) \geq n \geq \sum_{j=1}^{m} \max(Q, O_j) \end{cases}$$

Results for a scheduling problem

- m tasks by day, m workers, and n days;
- AllDifferent for each day (all tasks are performed by distinct workers);
- AtMostAllBalance for each worker (the workload of each worker is balanced over all tasks);
- Random unavailability to make the instances more real (25 instances for each couple (n, m));
- Minimization of the largest balance over workers.

m	n	Decomp. #	Decomp. B	Decomp. Time	Decomp. Bkt	Implied #	Implied B	Implied Time	Implied Bkt	Implied+ #	Implied+ B	Implied+ Time	Implied+ Bkt	Balance* #	Balance* B	Balance* Time	Balance* Bkt
6	16	8	1.92	6634	87398	**25**	**1.88**	37	472	**25**	**1.88**	35	423	**25**	**1.88**	**33**	**260**
6	17	11	**2.16**	60637	1073765	**25**	**2.16**	78	1123	**25**	**2.16**	63	877	**25**	**2.16**	**36**	**249**
6	18	16	3.2	8869	166146	**25**	**1.84**	127	1903	**25**	**1.84**	114	1617	**25**	**1.84**	**36**	**279**
6	19	8	3.24	106003	1352600	**25**	**2.64**	607	6983	**25**	**2.64**	504	6923	**25**	**2.64**	**61**	**408**
6	20	7	3.04	2302	27839	**25**	**2.80**	910	10027	**25**	**2.80**	734	8221	**25**	**2.80**	**169**	**1085**
7	16	6	**1.44**	32540	476847	**25**	**1.44**	2361	29767	**25**	**1.44**	2112	28382	**25**	**1.44**	**1828**	**12383**
7	17	9	2.04	159790	1542016	**25**	**1.96**	8416	90680	**25**	**1.96**	6697	72236	**25**	**1.96**	**1576**	**9378**
7	18	3	2.36	135580	1439674	22	1.76	19432	236671	22	1.76	14300	183069	**24**	**1.68**	13920	90665
7	19	4	2.04	80636	804503	22	1.88	21981	230327	22	1.88	13840	151262	**23**	**1.76**	6378	36822
7	20	2	2.72	25779	290430	23	1.56	46267	600434	**24**	**1.52**	55260	715789	23	1.68	18772	116406
8	16	8	2.12	128618	2109236	22	0.92	17420	231594	23	0.72	34216	462257	**25**	**0.44**	3797	14999
8	17	3	1.84	154183	1271700	21	1.68	55193	716866	21	1.68	49859	689151	**25**	**1.28**	12900	68059
8	18	1	1.76	4033	35971	15	1.56	56785	542326	**16**	**1.52**	84438	745177	**16**	1.56	5264	15636
8	19	2	2.12	176092	1675776	**24**	**1.40**	64074	665200	**24**	**1.40**	51899	544990	**24**	**1.40**	**31092**	**201842**
8	20	2	5.84	242901	2082063	11	2.76	51041	468643	11	2.68	35712	316148	**15**	**2.32**	12654	52741

Conclusion

1. We proved that achieving DC on Balance and AtMostBalance is NP-hard.
2. We introduced decompositions that perform well in practice.
3. We proposed a filtering algorithm for AtMostAllBalance that
   a) achieves domain consistency;
   b) runs in polynomial time;
   c) performs better than the decompositions.

Emilie Picard-Cantin, PhD student emilie.picard-cantin.1@ulaval.ca