Supplementary material: A constructive branch-and-bound algorithm for the project duration problem with partially renewable resources and general temporal constraints

Journal of Scheduling

Kai Watermeyer · Jürgen Zimmermann

Clausthal University of Technology
Institute of Management and Economics
Operations Research Group
Julius-Albert-Str. 2, 38678 Clausthal-Zellerfeld, Germany

Corresponding author: Kai Watermeyer
E-mail: kai.watermeyer@tu-clausthal.de
Jürgen Zimmermann
E-mail: juergen.zimmermann@tu-clausthal.de

This supplementary document provides a comparison between the constructive branch-and-bound algorithm (CBB) from the associated paper with the relaxation-based branch-and-bound algorithm (RBB) by Watermeyer and Zimmermann (2020). In contrast to the computational studies in the paper, both branch-and-bound algorithms (BnB) are conducted with only one setting over all instances of benchmark set UBOπ. Furthermore, this document contains complementary results for the devised serial schedule-generation scheme (SGS).

Tables 2 and 4, which are taken from the corresponding papers, summarize the settings that have been used for both BnB depending on the instance size. For the description of the symbols we refer the reader to the associated papers. In Tables 2 and 4, the different settings are marked by numbers ([1], [2], [3], [4], [5]) that are used in the evaluation tables to identify the settings.

Table 1 shows the results of the experimental performance analysis that has been conducted with a time limit of 300 seconds, where setting [6] of CBB corresponds to setting [5] with a so-called warm-up phase (noted by CBB+W in the paper). Tables 3 and 5 display the results for each test set UBO$n\pi$ with $n = 10, 20, 50, 100, 200$ real activities. The measures are described in the associated paper of this document.

First of all, from Tables 3 and 5 it can be seen that the performance of each setting is highly dependent on the instance size. As a rule, it can be established that settings that show a reasonable performance for small instances are rather not suited for greater instances and vice versa. Table 1 shows in addition that settings that perform well on small instances tend to solve more instances, while settings with a better performance for greater instances are able to determine the solution status for much more instances.

		#nTriv	#opt	#feas	#inf	#unk	Δ^{lb} (%)	$\varnothing^{\mathrm{cpu}}$ (s)	$\varnothing^{\mathrm{cpu}}_{\mathrm{opt}}$ (s)	$\varnothing^{\mathrm{cpu}}_{\mathrm{inf}}$ (s)
CBB	[1]	2791	1316	1613	205	973	173.12	139.083	5.237	1.405
	[2]		1371	2009	205	577	149.74	134.526	7.861	0.899
	[3]		1261	1880	203	708	165.91	145.862	6.692	2.767
	[4]		1352	2456	200	135	116.19	137.105	7.802	2.060
	[5]		1353	2523	200	68	111.52	137.015	7.708	2.905
	[6]		1353	2541	200	50	111.04	137.792	9.209	3.578
RBB	[1]	2791	1343	1581	203	1007	168.85	137.145	6.623	1.859
	[2]		1313	1669	203	919	173.17	140.753	7.611	1.716
	[3]		1110	2325	181	285	142.15	163.629	6.024	0.010
	[4]		1149	2526	181	84	114.95	160.850	9.252	0.009
	[5]		1120	2525	181	85	114.96	163.570	8.501	0.010

Table 1 Performance of CBB and RBB over all instances of benchmark set UBOπ (300 s)

Finally, the comparison between the results of CBB and RBB in Table 1 reveals that CBB dominates RBB if only one setting is used over all instances of benchmark set UBOπ. In particular, setting [6] for CBB can be identified as the best setting that is able to determine the solution status for the greatest number of instances, while the difference to the highest number of solved instances by setting [2] is rather low.

	UBO10π [1]	UBO20π [2]	UBO50π [3]	UBO100π [4]	UBO200π [5]
Traversing strategy	DFS	SPS [2 s]	SPS [5 s]	SPS [5 s]	SPS [5 s]
Branching strategy	$\bar{C}$, MRC(max)	$\bar{C}$, MRC(max)	$\prec_{\mathrm{D}}$, $\mathrm{STd}^{\mathrm{I}}$(min)	$\prec_{\mathrm{C}}$, PF(max)	$\prec_{\mathrm{C}}$, PF(max)
Generation strategy	restr-LT [10]	restr-LT [10]	restr-PV [5]	restr-PV [15]	restr-PV [15]
Ordering strategy	LB(min)	LB(min)	LB(min)	LB(min)	LB(min)
Consistency tests	γ_B^∞	γ_B^∞, $\gamma_D^1[\mathcal{R}_i,2]$	γ_B^∞, $\gamma_W^1[\mathcal{R}_i]$	γ_B^∞, $\gamma_W^1[\mathcal{R}_i,2]$	γ_B^∞
Lower bound	LBD^π	LBD^π	LBD^π	$LB0^\pi$	$LB0^\pi$
Pruning techniques	UPT	UPT	UPT+ULT	UPT+ULT	UPT+ULT

Table 2 Settings of CBB for the performance analysis on benchmark set UBOπ dependent on the instance size

		#nTriv	#opt	#feas	#inf	#unk	Δ^{lb} (%)	$\varnothing^{\mathrm{cpu}}$ (s)	$\varnothing^{\mathrm{cpu}}_{\mathrm{opt}}$ (s)	$\varnothing^{\mathrm{cpu}}_{\mathrm{inf}}$ (s)
UBO10π	[1]	693	534	534	159	0	53.43	0.065	0.067	0.056
	[2]		534	534	159	0	53.43	0.076	0.079	0.063
	[3]		534	534	159	0	53.43	0.140	0.163	0.063
	[4]		534	534	159	0	53.43	0.328	0.423	0.008
	[5]		534	534	159	0	53.43	0.410	0.512	0.066
	[6]		534	534	159	0	53.43	0.774	0.902	0.343
UBO20π	[1]	621	532	575	40	6	65.77	29.458	6.708	0.618
	[2]		537	581	40	0	64.67	28.086	7.846	0.702
	[3]		482	580	39	2	65.16	54.821	7.523	10.707
	[4]		476	580	36	5	65.98	59.072	7.835	7.062
	[5]		473	580	36	5	66.11	61.108	8.596	7.835
	[6]		474	580	36	5	66.09	62.242	10.558	9.655
UBO50π	[1]	527	161	326	6	195	119.24	209.408	13.065	42.412
	[2]		183	432	6	89	103.78	197.638	14.256	24.356
	[3]		183	491	5	31	88.16	198.016	13.774	26.827
	[4]		172	489	5	33	90.03	204.891	16.402	31.301
	[5]		167	489	5	33	90.20	207.570	15.576	57.677
	[6]		166	489	5	33	90.20	207.787	14.399	62.711
UBO100π	[1]	484	58	128	0	356	290.71	266.203	17.972	–
	[2]		78	302	0	182	237.99	257.415	35.754	–
	[3]		62	275	0	209	242.46	266.125	35.559	–
	[4]		85	472	0	12	168.68	249.827	14.307	–
	[5]		84	471	0	13	168.92	250.723	16.073	–
	[6]		84	472	0	12	168.87	251.274	19.244	–
UBO200π	[1]	466	31	50	0	416	432.97	280.345	4.540	–
	[2]		39	160	0	306	366.66	277.304	28.812	–
	[3]		0	0	0	466	475.85	300.000	–	–
	[4]		85	381	0	85	251.50	250.763	30.066	–
	[5]		95	449	0	17	222.93	243.429	22.502	–
	[6]		95	466	0	0	220.09	245.209	31.234	–

Table 3 Performance of CBB on benchmark set UBOπ (300 s)

	UBO10π [1]	UBO20π [2]	UBO50π [3]	UBO100π [4]	UBO200π [5]
Traversing strategy	DFS	SPS$^+$ [2 s]	SPS$^+$ [5 s]	SPS$^+$ [15 s]	SPS$^+$ [15 s]
Branching strategy	NCA(min)	NCA(min)	DST(max)	DST$^{\mathrm{I}}$(max)	DST$^{\mathrm{I}}$(max)
Generation strategy	all	all	all	restrCL [10]	restrCL [10]
Ordering strategy	LB(min)	LB(min)	LB-DST(min)	LB-DST$^{\mathrm{I}}$(min)	LB-DST$^{\mathrm{I}}$(min)
Consistency tests	γ_B^∞, $\gamma_W^1[\mathcal{R}_i,10]$	γ_B^∞, $\gamma_W^1[\mathcal{R}_i,10]$	γ_B^∞, $\gamma_D^1[\mathcal{R}_i]$	γ_B^∞, $\gamma_D^1[\mathcal{R}_i,5]$	γ_B^∞
Lower bound	LBD^π	LBD^π	LBD^π	$LB0^\pi$	$LB0^\pi$
Partitioning	x	x	–	–	–

Table 4 Settings of RBB for the performance analysis on benchmark set UBOπ dependent on the instance size

Note. From "A branch-and-bound procedure for the resource-constrained project scheduling problem with partially renewable resources and general temporal constraints" by K. Watermeyer and J. Zimmermann, 2020, OR Spectrum, 42(2), p. 456 (https://doi.org/10.1007/s00291-020-00583-z). CC BY 4.0 (https://creativecommons.org/licenses/by/4.0/).

		#nTriv	#opt	#feas	#inf	#unk	Δ^{lb} (%)	$\varnothing^{\mathrm{cpu}}$ (s)	$\varnothing^{\mathrm{cpu}}_{\mathrm{opt}}$ (s)	$\varnothing^{\mathrm{cpu}}_{\mathrm{inf}}$ (s)
UBO10π	[1]	693	534	534	159	0	53.43	0.032	0.040	0.004
	[2]		534	534	159	0	53.43	0.035	0.044	0.004
	[3]		508	534	157	2	53.62	14.084	2.676	0.005
	[4]		501	534	157	2	53.75	17.804	3.668	0.004
	[5]		493	534	157	2	53.71	21.113	3.510	0.004
UBO20π	[1]	621	497	560	40	21	67.35	48.082	8.671	8.733
	[2]		500	578	40	3	65.09	46.149	8.076	8.006
	[3]		361	578	21	22	69.51	117.381	3.306	0.011
	[4]		351	576	21	24	70.45	124.690	7.783	0.010
	[5]		338	575	21	25	70.46	130.852	7.866	0.012
UBO50π	[1]	527	174	255	4	268	128.33	201.974	9.843	6.869
	[2]		188	348	4	175	112.44	195.959	14.591	6.852
	[3]		145	486	3	38	95.49	217.958	8.022	0.279
	[4]		136	486	3	38	96.32	223.597	10.550	0.286
	[5]		133	484	3	40	96.57	225.069	9.852	0.292
UBO100π	[1]	484	78	127	0	357	286.91	253.566	11.869	–
	[2]		73	164	0	320	285.48	258.908	27.555	–
	[3]		72	386	0	98	226.00	260.174	32.279	–
	[4]		79	465	0	19	174.30	254.409	20.681	–
	[5]		77	466	0	18	174.74	255.554	20.623	–
UBO200π	[1]	466	60	105	0	361	398.94	265.503	32.076	–
	[2]		18	45	0	421	447.29	290.938	65.395	–
	[3]		24	341	0	125	336.24	285.936	26.915	–
	[4]		82	465	0	1	224.73	253.631	36.489	–
	[5]		79	466	0	0	224.03	253.934	28.271	–

Table 5 Performance of RBB on benchmark set UBOπ (300 s)

Table 6 provides the results of the devised SGS in the associated paper for the Böttcher and SAV benchmark set. The results in Table 6 were obtained by using the priority rules LST and ST with ext = min over all test sets of the Böttcher and SAV benchmark set, respectively. The calculation of T_i as well as the application of both improving techniques result in a better performance on both benchmark sets. A closer look on the results even show that this is also the case for each single test set. The best variant of SGS (+UPT) shows a reasonable performance. Only for the Böttcher benchmark set, SGS is not able to determine a feasible solution for each instance that was feasibly solved by CBB. It is worth mentioning that SGS with all improving techniques clearly outperforms the results of the SGS by Schirmer (1999) and Böttcher et al. (1999) on the Schirmer instances (J10π, J20π, J30π, J40π) in terms of the number of feasibly solved instances as reported in Table 17.2 in Schirmer (1999, Sect. 17). For this comparison we have conducted computational experiments on the Schirmer instances by including all trivial instances in accordance with the investigations in Schirmer (1999). Thereby, SGS was able to determine a feasible solution for each trivial instance.

			$Z = 100$				$Z = 1000$			
		#inst	#nfeas	Δ^{lb} (%)	Δ^{CBB} (%)	$\varnothing^{\mathrm{cpu}}$ (s)	#nfeas	Δ^{lb} (%)	Δ^{CBB} (%)	$\varnothing^{\mathrm{cpu}}$ (s)
Böttcher	SGS(Θ_i)	1399	86	46.76	2.04	0.074	54	44.72	0.66	0.113
	SGS(T_i)		71	45.08	0.68	0.075	49	44.25	0.26	0.117
	+RB		53	44.12	0.06	0.072	45	43.90	0.03	0.104
	+UPT		43	43.91	0.05	0.072	37	43.76	0.09	0.104
	CBB		22	43.28	0.00	6.525				
SAV	SGS(Θ_i)	2553	40	19.23	9.95	0.178	11	10.06	2.17	0.476
	SGS(T_i)		29	16.60	7.72	0.166	10	9.82	1.97	0.482
	+RB		11	14.83	6.28	0.164	4	9.11	1.42	0.467
	+UPT		0	10.86	3.05	0.117	0	7.76	0.35	0.331
	CBB		0	7.34	0.00	18.834				

Table 6 Performance of SGS on the Böttcher and SAV benchmark set