CERN

Automated scheduling optimisation

Development status of the “Supercycle Optimiser” program

Sandy Easton (BE-OP-PS)

16th May 2023

Acknowledgements

Special thanks to

- Hannes Pahl (BE-OP-PS)
  - Made the whole program apart from the computation backend
    - PyQt UI for inputting scheduling requests
    - Interfaced program to Timing Server using LIC API (from BE-CSS)
    - Launching the Java algorithm and displaying the results
  - Excellent help and support

And also thanks to

- Tibor Bukovics (BE-OP-PSB) & Andrea Callia (BE-OP-LHC)

  Software engineering support

- Jean-Charles Tournier (ICS) & Bertrand Lefort (BE-OP-AD)

  Useful algorithm-whiteboard sessions

- George Melvin & Alex Paulin (UC Berkeley), Spencer Gessner (SLAC) & Solve Slettebak (ESS)

  Theoretical discussions and encouragement

- Alex Scheinker (LANL), Andrew Wooff (Napier), & Mike McKerns (Caltech/LANL)

  Broader scope of algorithms, and uniqueness of new functionality

- Nico Madysa, Greg Kruk, Verena Kain, Zsolt Kovari & Andrejz Dvorak (BE-CSS)

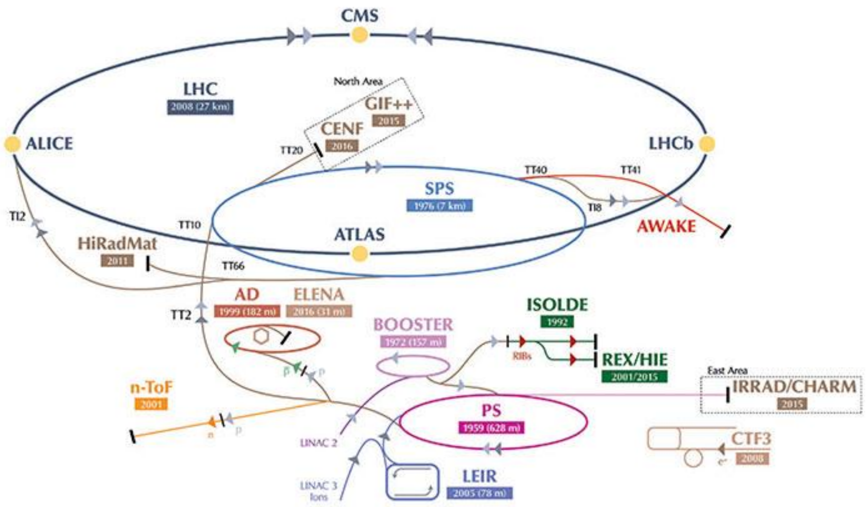
  Programming assistance and raising project visibility

And also thanks to

- Bettina Mikulecs (BE-OP-PS) & Frank Tecker (CAS)
  - Long suffering supervisors ☺
- Eva Maria Gonzalez Garcia (BE-OP-PS)
  - Technical student assisting project 2020-2021
- Prof. Andrew Ranicki (1948-2018) (Edinburgh)
  - Former advisor and supervisor
- and too many operators to mention!

Overview

Scheduling beams



- CERN's accelerators are a chain
  - Beams come from only two sources

- Trafficing beams inefficiently wastes experiment-time
  - Geneva traffic vs. Geneva airport….
  - Scheduling bottlenecks -> Fewer beams/minute
  - Not an insignifcant loss

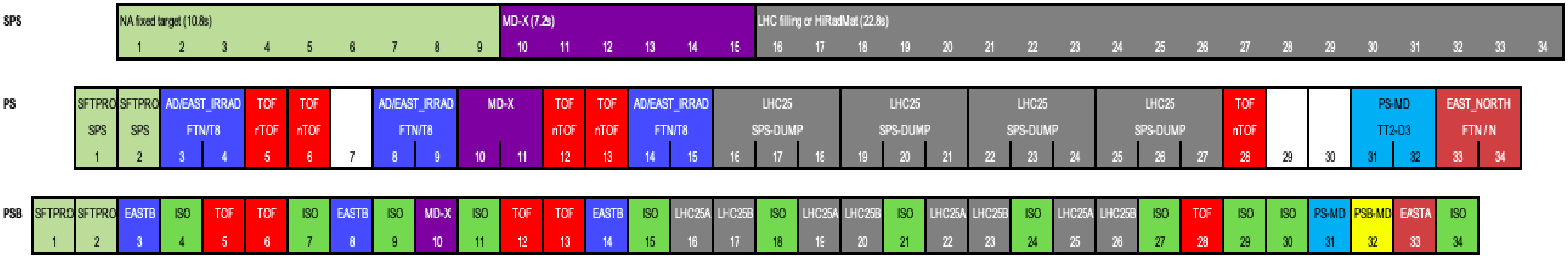
- The schedule is adjusted during real-time operations
  - Every time an experiment's requirement changes
  - 20 – 80 times per day

- At any given time, this scheduling task is handled by the accelerator operators in the CCC
  - Knowledge of which beams are to be taken
  - Knowledge of all scheduling constraints
    - How many to whom?
    - User requirements, hardware constraints, etc.
  - Knowledge of which users are given highest priorities

- The operators agree upon a new schedule at every change
  - A.k.a. "change the * *Supercycle*"
- This schedule is used until the next change in a user requirement

** "Supercycle" = the schedule of beams to be created by the accelerator chain*

The Supercycle



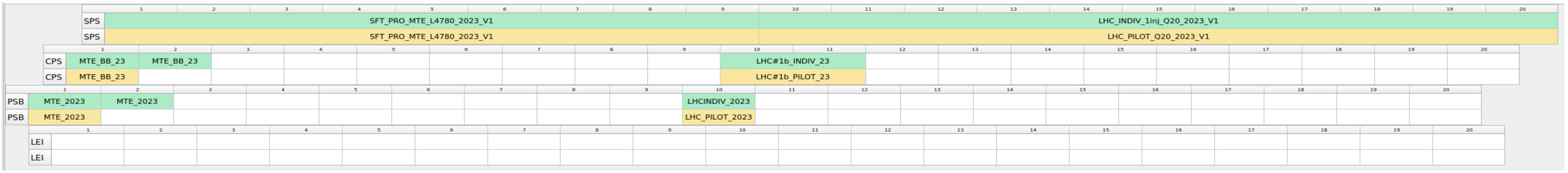
SPS	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
	NA fixed target (10.8s)									MD-X (7.2s)						LHC filling or HiRadMat (22.8s)																		

PS	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
	SFTPRO SPS	SFTPRO SPS	AD/EAST_IRRAD FTN/T8		TOF nTOF	TOF nTOF		AD/EAST_IRRAD FTN/T8		MD-X		TOF nTOF	TOF nTOF	AD/EAST_IRRAD FTN/T8		LHC25 SPS-DUMP			LHC25 SPS-DUMP			LHC25 SPS-DUMP			LHC25 SPS-DUMP			TOF nTOF			PS-MD TT2-D3		EAST_NORTH FTN / N	

PSB	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
	SFTPRO	SFTPRO	EASTB	ISO	TOF	TOF	ISO	EASTB	ISO	MD-X	ISO	TOF	TOF	EASTB	ISO	LHC25A	LHC25B	ISO	LHC25A	LHC25B	ISO	LHC25A	LHC25B	ISO	LHC25A	LHC25B	ISO	TOF	ISO	ISO	PS-MD	PSB-MD	EASTA	ISO

- Observe
  - Green beams only using the booster (PSB user)
  - Red blue and brown beams to PS users
    - Pass through PSB & PS
  - Teal, Purple and Grey to SPS users
    - PSB, PS & SPS

  - And this still omits the LEIR…!
  - Behold, a real supercycle from the Timing App Suite (existing Supercycle edition software)



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
SPS	SFT_PRO_MTE_L4780_2023_V1									LHC_INDIV_1inj_Q20_2023_V1										
SPS	SFT_PRO_MTE_L4780_2023_V1									LHC_PILOT_Q20_2023_V1										

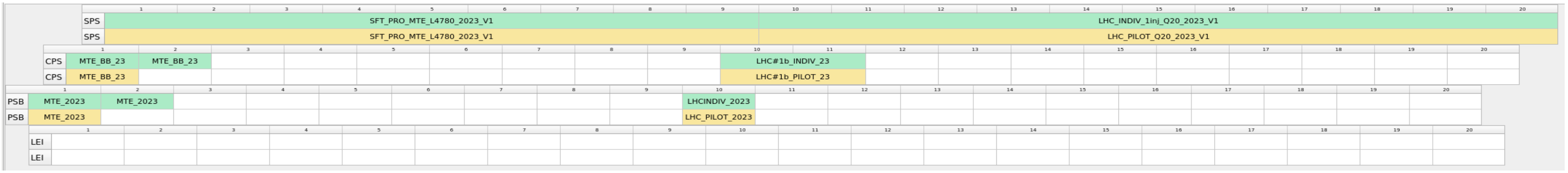
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
CPS	MTE_BB_23	MTE_BB_23								LHC#1b_INDIV_23										
CPS	MTE_BB_23									LHC#1b_PILOT_23										

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
PSB	MTE_2023	MTE_2023								LHCINDIV_2023										
PSB	MTE_2023									LHC_PILOT_2023										

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
LEI																				
LEI																				

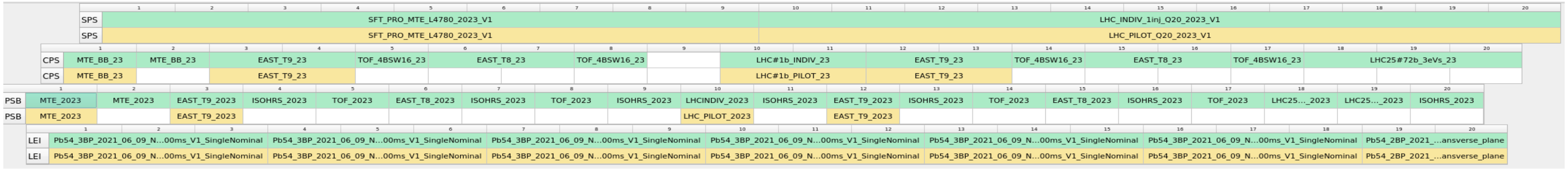
The Supercycle

- Indeed, this is the skeleton when making a supercycle
  - The beams to the SPS users



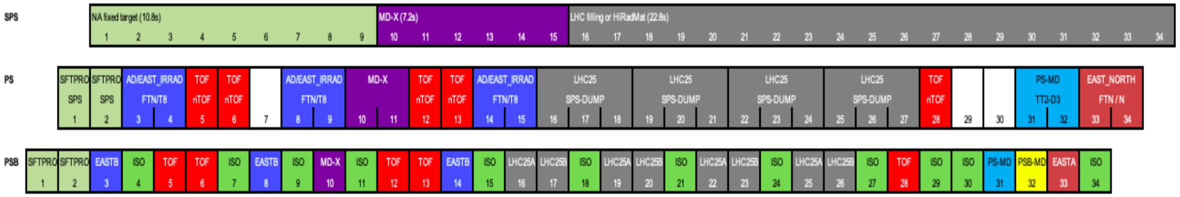
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
SPS	SFT_PRO_MTE_L4780_2023_V1									LHC_INDIV_1inj_Q20_2023_V1										
SPS	SFT_PRO_MTE_L4780_2023_V1									LHC_PILOT_Q20_2023_V1										
CPS	MTE_BB_23	MTE_BB_23								LHC#1b_INDIV_23										
CPS	MTE_BB_23									LHC#1b_PILOT_23										
PSB	MTE_2023	MTE_2023								LHCINDIV_2023										
PSB	MTE_2023									LHC_PILOT_2023										
LEI																				
LEI																				

- CPS beams are then placed in the available white space
  - ISOLDE, EAST, AD, TOF
  - MD users
  - Beams under adjustment



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
SPS	SFT_PRO_MTE_L4780_2023_V1									LHC_INDIV_1inj_Q20_2023_V1										
SPS	SFT_PRO_MTE_L4780_2023_V1									LHC_PILOT_Q20_2023_V1										
CPS	MTE_BB_23	MTE_BB_23	EAST_T9_23		TOF_4BSW16_23	EAST_T8_23		TOF_4BSW16_23		LHC#1b_INDIV_23		EAST_T9_23		TOF_4BSW16_23	EAST_T8_23		TOF_4BSW16_23	LHC25#72b_3eVs_23		
CPS	MTE_BB_23		EAST_T9_23							LHC#1b_PILOT_23		EAST_T9_23								
PSB	MTE_2023	MTE_2023	EAST_T9_2023	ISOHRS_2023	TOF_2023	EAST_T8_2023	ISOHRS_2023	TOF_2023	ISOHRS_2023	LHCINDIV_2023	ISOHRS_2023	EAST_T9_2023	ISOHRS_2023	TOF_2023	EAST_T8_2023	ISOHRS_2023	TOF_2023	LHC25..._2023	LHC25..._2023	ISOHRS_2023
PSB	MTE_2023		EAST_T9_2023							LHC_PILOT_2023		EAST_T9_2023								
LEI	Pb54_3BP_2021_06_09_N...00ms_V1_SingleNominal			Pb54_3BP_2021_06_09_N...00ms_V1_SingleNominal			Pb54_3BP_2021_06_09_N...00ms_V1_SingleNominal			Pb54_3BP_2021_06_09_N...00ms_V1_SingleNominal			Pb54_3BP_2021_06_09_N...00ms_V1_SingleNominal			Pb54_3BP_2021_06_09_N...00ms_V1_SingleNominal			Pb54_2BP_2021_...ansverse_plane	
LEI	Pb54_3BP_2021_06_09_N...00ms_V1_SingleNominal			Pb54_3BP_2021_06_09_N...00ms_V1_SingleNominal			Pb54_3BP_2021_06_09_N...00ms_V1_SingleNominal			Pb54_3BP_2021_06_09_N...00ms_V1_SingleNominal			Pb54_3BP_2021_06_09_N...00ms_V1_SingleNominal			Pb54_3BP_2021_06_09_N...00ms_V1_SingleNominal			Pb54_2BP_2021_...ansverse_plane	

- The individual beams are hard to make outthere, don’t worry it’s just

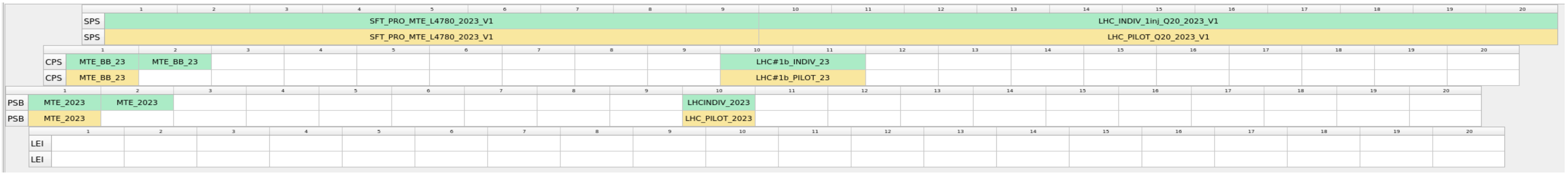


	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
SPS	NA fixed target (10.8s)									MD-X (7.2s)						LHC filling or HiRadMat (22.8s)	
PS	SFTPRO SPS	SFTPRO SPS	AD/EAST_IRRAD FTN/T8		TOF nTOF	TOF nTOF		AD/EAST_IRRAD FTN/T8		MD-X		TOF nTOF	TOF nTOF	AD/EAST_IRRAD FTN/T8		LHC25 SPS-DUMP	
PSB	SFTPRO	SFTPRO	EASTB	ISO	TOF	TOF	ISO	EASTB	ISO	MD-X	ISO	TOF	TOF	EASTB	ISO	LHC25A	LHC25B

	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
SPS																	
PS		LHC25 SPS-DUMP			LHC25 SPS-DUMP			LHC25 SPS-DUMP			TOF nTOF			PS-MD TT2-D3		EAST_NORTH FTN / N	
PSB	ISO	LHC25A	LHC25B	ISO	LHC25A	LHC25B	ISO	LHC25A	LHC25B	ISO	TOF	ISO	ISO	PS-MD	PSB-MD	EASTA	ISO

Changing the supercycle

- This schedule creation is performed manually by the CPS operators



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
SPS	SFT_PRO_MTE_L4780_2023_V1									LHC_INDIV_1inj_Q20_2023_V1										
SPS	SFT_PRO_MTE_L4780_2023_V1									LHC_PILOT_Q20_2023_V1										

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
CPS	MTE_BB_23	MTE_BB_23								LHC#1b_INDIV_23										
CPS	MTE_BB_23									LHC#1b_PILOT_23										

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
PSB	MTE_2023	MTE_2023								LHCINDIV_2023										
PSB	MTE_2023									LHC_PILOT_2023										

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
LEI																				
LEI																				

Which users in each accelerator?

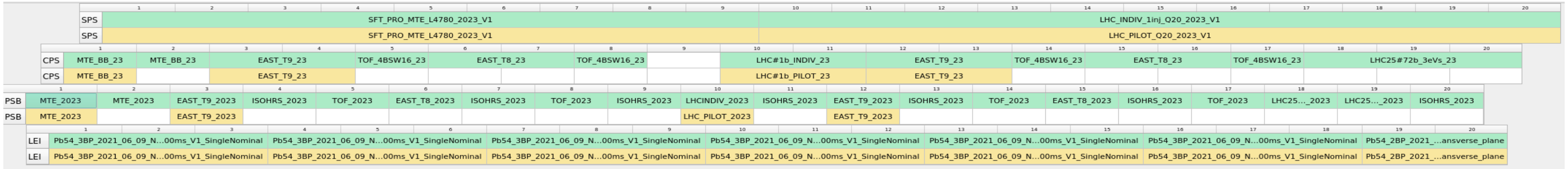
How many should there be of each user?

Repetition rates?

Positioning constraints?

Who has priority for free space?

Hardware constraints?



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
SPS	SFT_PRO_MTE_L4780_2023_V1									LHC_INDIV_1inj_Q20_2023_V1										
SPS	SFT_PRO_MTE_L4780_2023_V1									LHC_PILOT_Q20_2023_V1										

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
CPS	MTE_BB_23	MTE_BB_23	EAST_T9_23		TOF_4BSW16_23	EAST_T8_23		TOF_4BSW16_23		LHC#1b_INDIV_23		EAST_T9_23		TOF_4BSW16_23	EAST_T8_23		TOF_4BSW16_23	LHC25#72b_3eVs_23		
CPS	MTE_BB_23		EAST_T9_23							LHC#1b_PILOT_23		EAST_T9_23								

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
PSB	MTE_2023	MTE_2023	EAST_T9_2023	ISOHRS_2023	TOF_2023	EAST_T8_2023	ISOHRS_2023	TOF_2023	ISOHRS_2023	LHCINDIV_2023	ISOHRS_2023	EAST_T9_2023	ISOHRS_2023	TOF_2023	EAST_T8_2023	ISOHRS_2023	TOF_2023	LHC25..._2023	LHC25..._2023	ISOHRS_2023
PSB	MTE_2023		EAST_T9_2023							LHC_PILOT_2023		EAST_T9_2023								

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
LEI	Pb54_3BP_2021_06_09_N...00ms_V1_SingleNominal			Pb54_3BP_2021_06_09_N...00ms_V1_SingleNominal			Pb54_3BP_2021_06_09_N...00ms_V1_SingleNominal			Pb54_3BP_2021_06_09_N...00ms_V1_SingleNominal			Pb54_3BP_2021_06_09_N...00ms_V1_SingleNominal			Pb54_3BP_2021_06_09_N...00ms_V1_SingleNominal			Pb54_2BP_2021_...ansverse_plane	
LEI	Pb54_3BP_2021_06_09_N...00ms_V1_SingleNominal			Pb54_3BP_2021_06_09_N...00ms_V1_SingleNominal			Pb54_3BP_2021_06_09_N...00ms_V1_SingleNominal			Pb54_3BP_2021_06_09_N...00ms_V1_SingleNominal			Pb54_3BP_2021_06_09_N...00ms_V1_SingleNominal			Pb54_3BP_2021_06_09_N...00ms_V1_SingleNominal			Pb54_2BP_2021_...ansverse_plane	

- It takes 1-5 minutes to edit a sequence, depending on complexity
- All operators should verify it
- 20-80 times per day

Scheduling

- Typically supercycles range from 20-40 blocks* long
    - Occasionally higher
    - As high as 80 BPs!

- The supercycle is "cyclic"
    - It repeats cyclically until changed
    - This is surprisingly tricky for algorithms to deal with…!

- Beams can each take up between 1-3 blocks in the CPS accelerators
    - Beams can fit under each other
    - Parallel scheduling
        1. Baby Terrapin
        2. Similarities to other scheduling problems (KT plug)

- Beams can need to be regularly spaced etc. (a.k.a. "Constraints")

- This presents a sizable scheduling challenge!

* The supercycle is quantised into blocks of 1.2 seconds, a.k.a. "basic periods" or BPs

Why is automation necessary

- This task takes up considerable operator time
    - CPS operators put most time into this task
    - Considerable distraction for administrative task

- Estimated 30 minutes to 3 hours cumulative operator time per day
    - Requires communal input from operators of all accelerators

- Existing system is effective for manual scheduling like this
    - Manually edit schedules and send it to Timing hardware
    - ”Locking” system to prevent concurrency mistakes
    - Requires plenty communication and good timing

- However, manual scheduling itself is
    - Laborious
    - At least occasionally sub-optimal
    - Open to some errors

Why is optimisation important?

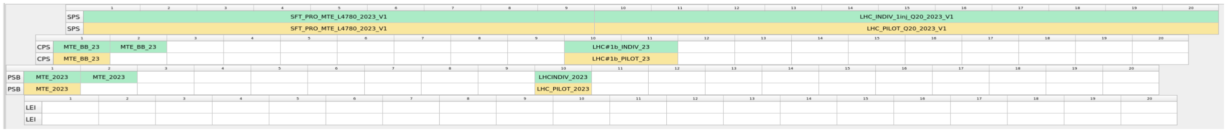
- Example
  - Suppose it is found that a rearrangement of the Supercycle can allow an extra instance to a user
  - This user had two instances, but now gets three
  - 50% increase in rate of data taking for that user

- Example
  - Consider a Supercycle of length 25 BP
  - It is found to be suboptimal by a single BP in one accelerator going unnecessarily unused
  - Akin to a 4% loss in that accelerator’s uptime
    - AFT….? ☺
  - 2% for a 50 BP supercycle

- Small scheduling inefficiencies are a serious bottleneck for operations
  - Small oversights can cost a lot

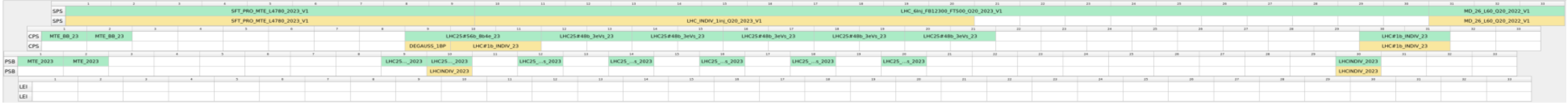
Schedule boundaries

1. Suppose the SPS changes its skeleton request from



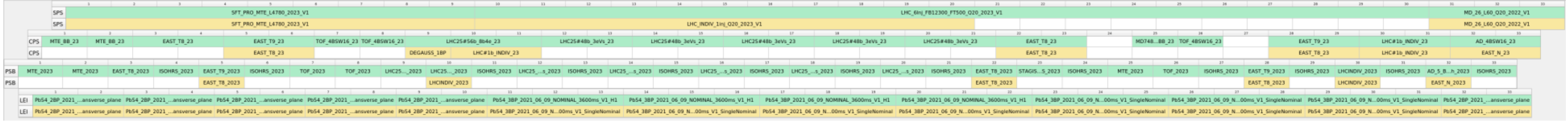
SPS SFT_PRO_MTE_L4780_2023_V1 | LHC_INDIV_1inj_Q20_2023_V1
SPS SFT_PRO_MTE_L4780_2023_V1 | LHC_PILOT_Q20_2023_V1
CPS MTE_BB_23 | MTE_BB_23 | LHC#1b_INDIV_23
CPS MTE_BB_23 | LHC#1b_PILOT_23
PSB MTE_2023 | MTE_2023 | LHCINDIV_2023
PSB MTE_2023 | LHC_PILOT_2023
LEI
LEI

2. To



SPS SFT_PRO_MTE_L4780_2023_V1 | LHC_6inj_FB12300_FT500_Q20_2023_V1 | MD_26_L60_Q20_2022_V1
SPS SFT_PRO_MTE_L4780_2023_V1 | LHC_INDIV_1inj_Q20_2023_V1 | MD_26_L60_Q20_2022_V1
CPS MTE_BB_23 | MTE_BB_23 | LHC25#56b_8b4e_23 | LHC25#48b_3eVs_23 | LHC25#48b_3eVs_23 | LHC25#48b_3eVs_23 | LHC25#48b_3eVs_23 | LHC25#48b_3eVs_23 | LHC#1b_INDIV_23
CPS DEGAUSS_1BP | LHC#1b_INDIV_23 | LHC#1b_INDIV_23
PSB MTE_2023 | MTE_2023 | LHC25..._2023 | LHC25..._2023 | LHC25_...s_2023 | LHC25_...s_2023 | LHC25_...s_2023 | LHC25_...s_2023 | LHC25_...s_2023 | LHCINDIV_2023
PSB LHCINDIV_2023 | LHCINDIV_2023
LEI
LEI

3. The PS, booster and LEIR operators must all now fill this space
    - … in ”the same way” they did before
    - obey all their previous constraints and requests



SPS SFT_PRO_MTE_L4780_2023_V1 | LHC_6inj_FB12300_FT500_Q20_2023_V1 | MD_26_L60_Q20_2022_V1
SPS SFT_PRO_MTE_L4780_2023_V1 | LHC_INDIV_1inj_Q20_2023_V1 | MD_26_L60_Q20_2022_V1
CPS MTE_BB_23 | MTE_BB_23 | EAST_T8_23 | EAST_T9_23 | TOF_4BSW16_23 | TOF_4BSW16_23 | LHC25#56b_8b4e_23 | LHC25#48b_3eVs_23 | LHC25#48b_3eVs_23 | LHC25#48b_3eVs_23 | LHC25#48b_3eVs_23 | LHC25#48b_3eVs_23 | EAST_T8_23 | MD748...BB_23 | TOF_4BSW16_23 | EAST_T9_23 | LHC#1b_INDIV_23 | AD_4BSW16_23
CPS EAST_T8_23 | DEGAUSS_1BP | LHC#1b_INDIV_23 | EAST_T8_23 | EAST_T8_23 | LHC#1b_INDIV_23 | EAST_N_23
PSB MTE_2023 | MTE_2023 | EAST_T8_2023 | ISOHRS_2023 | EAST_T9_2023 | ISOHRS_2023 | TOF_2023 | TOF_2023 | LHC25..._2023 | LHC25..._2023 | ISOHRS_2023 | LHC25_...s_2023 | ISOHRS_2023 | LHC25_...s_2023 | ISOHRS_2023 | LHC25_...s_2023 | ISOHRS_2023 | LHC25_...s_2023 | ISOHRS_2023 | LHC25_...s_2023 | ISOHRS_2023 | EAST_T8_2023 | STAGIS...S_2023 | ISOHRS_2023 | MTE_2023 | TOF_2023 | ISOHRS_2023 | EAST_T9_2023 | ISOHRS_2023 | LHCINDIV_2023 | ISOHRS_2023 | AD_5_B...h_2023 | ISOHRS_2023
PSB EAST_T8_2023 | LHCINDIV_2023 | EAST_T8_2023 | EAST_T8_2023 | LHCINDIV_2023 | EAST_N_2023
LEI Pb54_2BP_2021_...ansverse_plane | Pb54_2BP_2021_...ansverse_plane | Pb54_2BP_2021_...ansverse_plane | Pb54_2BP_2021_...ansverse_plane | Pb54_2BP_2021_...ansverse_plane | Pb54_3BP_2021_06_09_NOMINAL_3600ms_V1_H1 | Pb54_3BP_2021_06_09_NOMINAL_3600ms_V1_H1 | Pb54_3BP_2021_06_09_NOMINAL_3600ms_V1_H1 | Pb54_3BP_2021_06_09_NOMINAL_3600ms_V1_H1 | Pb54_3BP_2021_06_09_N...00ms_V1_SingleNominal | Pb54_3BP_2021_06_09_N...00ms_V1_SingleNominal | Pb54_3BP_2021_06_09_N...00ms_V1_SingleNominal | Pb54_2BP_2021_...ansverse_plane
LEI Pb54_2BP_2021_...ansverse_plane | Pb54_2BP_2021_...ansverse_plane | Pb54_2BP_2021_...ansverse_plane | Pb54_2BP_2021_...ansverse_plane | Pb54_2BP_2021_...ansverse_plane | Pb54_3BP_2021_06_09_N...00ms_V1_SingleNominal | Pb54_3BP_2021_06_09_N...00ms_V1_SingleNominal | Pb54_3BP_2021_06_09_N...00ms_V1_SingleNominal | Pb54_3BP_2021_06_09_N...00ms_V1_SingleNominal | Pb54_3BP_2021_06_09_N...00ms_V1_SingleNominal | Pb54_3BP_2021_06_09_N...00ms_V1_SingleNominal | Pb54_3BP_2021_06_09_N...00ms_V1_SingleNominal | Pb54_2BP_2021_...ansverse_plane

4. This task must be performed whenever any operator has to change their request

Project aims

- *Automate the task of building supercycles*

- Explore and implement constraint-solving and optimisation algorithm
  - Crux of the project!

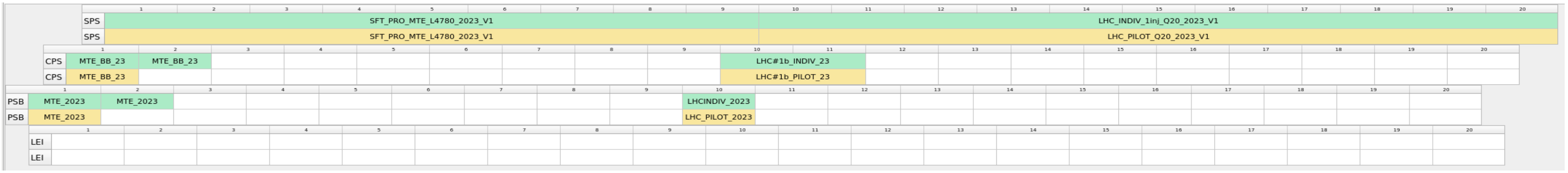
- Proof of concept for automation principle
  - Reduce scheduling requirements to UI input
  - Deliver maximally-efficient scheduling for near real-time use
    - Should be faster than a human as a general rule

- Produce tool for operational use in CCC
  - Interface optimiser to existing Supercycle editor software
  - Timing App Suite used for all viewing and driving tasks

Inputting a scheduling request

Inputting a scheduling request

1. SPS defines the “skeleton” supercycle
   - Usual method: Timing App Suite



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
SPS	SFT_PRO_MTE_L4780_2023_V1									LHC_INDIV_1inj_Q20_2023_V1										
SPS	SFT_PRO_MTE_L4780_2023_V1									LHC_PILOT_Q20_2023_V1										

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
CPS	MTE_BB_23	MTE_BB_23								LHC#1b_INDIV_23										
CPS	MTE_BB_23									LHC#1b_PILOT_23										

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
PSB	MTE_2023	MTE_2023								LHCINDIV_2023										
PSB	MTE_2023									LHC_PILOT_2023										

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
LEI																				
LEI																				

   - All BCDs are available to be read
   1. Demonstration of input and output BCDs for optimiser, in TAS & OptimiserUI
   2. Have prepared 3 inputs for us to use

N.B. keeping track of beams

1. Beams are named by concatenating the names of their constituent LSA cycles
2. Horible example
3. This is unwieldy, so here we make use of beam aliases
4. Picture of beam aliases

Demonstration of

- Importing an SPS fixed sequence
- A simple request
  - Using beam aliases*

What are we looking for?

We are looking for a Supercycle that is

1. Optimal
   - Produce as many usable beams per minute as possible from the complex
2. Valid
   - All constraints for all beams must be satisfied

Optimality with prioritising

1. If any extra space is available in the Supercycle, it is allocated to users according to their priority.
2. Every user has a minimum required precence in the Supercycle
3. Allocations above this need to be ranked for the optimiser to have an objective function
   - This makes an interesting discretised objective function to be dealt with….
4. It’s not pretty but it works…
   - Thanks be to Hannes!

Demonstration of

- Using the priority list
- Changing SPS sequence transparently

User Allocations

Demonstration of

- Maximum allocation
- Minimum allocation
  - Types of occurrences calculation

Types of occurences definition

- Existing
  - Fixed
    - MDs
  - x per y BPs
    - Constant-rate experiments
    - TOF, EAST, ISOLDE
  - Percentage
    - ISOLDE 37-40%

- Potential
  - “Equally spaced”
  - From user sources
    - ISOLDE current & current-per-shot
    - TOF radiation monitors

Output

1. If a solution is possible, output is a Supercycle that could be sent to the timing system
    1. With the allocation/runtime and the sequence
2. Picutre
3. If a solution is not possible, output describes why
    1. Constraints
    2. Request unsaisfiable

Demonstration of failures

- Constraints Unsatisfiable
- Allocations too high
- Max and min auto adjust
- No LEIRs
- Guidance on why requests were not possible

Positioning constraints

Demonstrations

1. Min
    1. Cyclicity
2. Max
    1. Max spacings buggy in hangover…
       (Implementation error)

Always Follow

- It is regularly requested that one beam directly follow a particular other
  - Typically for a constant hysteresis
- The optimiser must thus allow this to be requested

Demonstrations

- Always follows

Block placements

Some experiments (typically @ ISOLDE) ask for their shots to be grouped together

For example

- In groups of 2
- In groups of 2,3 or 4
- In groups of 8!

The optimiser must all thus be able to handle these kinds of requests

Demonstration

- Block placements
- Blocks with spacings

Spares

- Three use-cases pinpointed:
  1. AD/East scenario
  2. Comparison
     - Two beams to be alternated between using the Request
     - MD users
  3. Parasitic
     - Desperately hoping for an external condition…!

- Comparison type implemented
- AD/EAST with partial functionality
- Parasitic not implemented at all (minor)

Demonstration

- Beams in spare
- Combination demos

Planned but not yet implemented

- POPS RMS constraints
  - Limit what can be requested to not overload POPS
  - Requires numerical parameter
- AD wait-time optimiser
  - Figured-out but not implemented
- Occurrences calculation from external sources (ISOLDE, TOF)
- Decoupled LEIR
- Adding duplicate beams to dump
- User preferences
- Speed boosts

Demonstration

- General demo

Algorithm & Implementation

What is the algorithm doing?

- The algorithm (Terrapin) is basically a sequence of two algorithms:
  - First one (WGP) deals with positioning constraints
    - Creates a 3-D network containing all possible valid supercycles
  - Second one (Kiwanda) finds a supercycle with the best allocation to users
    - Finds a suitably “long” path in the 3-D network

- Terrapin optimises by gradually increasing the requested allocation
  - According to the priority list
  - Can be seen in the terminal output

Satisfying constraints

- There are numerous constraint-solvers available. In OP, we already use
  - a heuristic-repair algorithm for some constrained scheduling
    - Scheduling LHC hardware tests
  - CSP algorithms

- Here we must not just find a valid sequence
  - We need enough knowledge of the valid sequences to find our preferred one
- Most constraint-solving algorithms focus on *if* the constraints are solvable
- Unsuitable for mapping paths for optimisation

- WGP uses a “correct-by-construction” approach to map the space of supercycles
  - Directly builds all possible extensions, no need to check for violations.
  - Efficient construction and storage of entire schedule-space

Runtime

- Development: Functionality vs runtime

- The slowest part of the algorithm is the path-tracing part
- Astronomically large possibilities
    1. Current runtime
    2. Where can it be sped up?
        1. 2-3 orders of magnitude easily
        2. 4-6 orders reasonably easily
        3. More on complex problems

- Is quite involved work. Decision to be made

Next steps

Development and deployment

1. Broader plans from within BE
    - There is an perational need for this tool
    - Is On-the-fly possible

2. On the fly is actually easy
    - Should decide which route to take

Documentation

1. Algorithm
    1. Document for publication
2. Implementation
    1. Handover?
    2. Other uses

Image credit to @CaroCalendula

Questions

CERN