Pugh Matrix for Scheduler Trade Study

Glossary:		
	Weight	The relative importance of the requirement (from 0, i.e. least important, to 5, i.e the most important)
	QC	Queue Coordinator
	ILPs	Integer Linear Programs (Gurobi, which possible Multi-Objective Function Extension when stated)
	GAs	Genetic Algorithms (with possible Multi-Objective Evolutionary Algorithm Extension when stated)
	HCAs	Hill-Climbing based algorithms
	Spike	Spike algorithm for Hubble Scheduling
	GrMax	Bryan Miller's Greedy-Max algorithm

Requirements	**Weight**	**QCs**	**ILPs**	**GAs**	**HCAs**	**Spike**	**GrMax**	**Notes**
Schedule both sites together / separate	4	0	2	1	-1	-2	2	ILPs: This is done automatically through the constraints and requires no effort.
								GAs: Make sure chromosomes do not mix illegally. Observations that can be scheduled at both sites are problematic: may end up in chromosomes for both sites.
								HCAs: A solution is site-based; I'm not sure how we will translate this to multi-site support.
								Spike: Spike only offers support for a single site.
								GM: Generalized to any number of sites
Handle ToOs (including interrupts)	5	0	1	1	1	2	1	ILPs: Interrupts bypass the scheduler, the ToO is executed, and then the scheduler is rerun.
								GAs: Same as above.
								HCAs: Same as above.
								Spike: Spike was designed to work with ToOs.
								GM: Interrupts bypass this, then rerun automatically to get the schedule for the rest of the night
Use resource and telescope calendar	2	0	2	1	1	1	2	ILPs: Resource availability and telescope calendar can be represented at model build time before running the scheduler.
								GAs: Resource availability guaranteed in initial chromosomes. Ensuring operations do not result in illegal schedules requires checking each operation.
								HCAs: Resource availablility guaranteed in initial solution chosen. Remove neighbors from neighbourhood that violate illegal schedules.
								Spike: I think Spike offers some constraint planning for resource availability.
								GM: Should be able to handle easily as part of timing constraints, needs prototyping
Fix an observation in place	2	0	1	0	1	2	1	ILPs: Variables can be set to 1 when building the model to force observations to be scheduled at certain times.
								GAs: Hole set in initial chromosomes. Maintaining hole through operations might be difficult. (Idea: don't allow certain operations, or disallow any operations that touch the hole.)
								HCAs: Similar to the GA problem, although since the operations are easier, we can simpy ignore neighbohrs that move the fixed observation(s).
								Spike: It should be possible to schedule an observation at a specific point using timing constraints and high priority..
								GM: Should be possible, either with a timing constraint or by specifying the time that the scheduler considers
Schedule around a hole	1	0	1	-1	1	2	2	ILPs: Exclude variables that represent the hole's time slot from the model, or set them to 0. (Either will work.)
								GAs: Leave blank in initial chromosomes. Again, we must exercise more caution as mutators might move observations into the hole, so need to represent holes & reject bad moves.
								HCAs: This is largely the same as in the case of fixing an observation. We simply do not allow neighbors to contain something in vw/el
								Spike: Holes can be specified in the input files.
								GM: Should be possible, either with a timing constraint or by specifying the time that the scheduler considers
Use forecast information / condition statistics	4	0	2	1	1	0	2	ILPS: Only select valid observations for forecast / conditions. Will have to rerun if conditions change.
								GAs: Only select valid observations for forecast / conditions. Will have to rerun if conditions change.
								HCAs: Same as GAs.
								Spike: I do not see any support for forecasts. In he original problem, being for a space-based telescope, this was not an issue.
								GM: Should be able to have conditions constraints as a function of time, needs prototyping
Science priorities (variable: score changes on what scheduled)	5	0	-2	1	2	-2	2	ILPs: We cannot change the objective function as things are scheduled. Can rerun schedules with new priorities based on assumption that observation completes to prepare.
								GAs: The priorities can change dependent on what else is scheduled. Since initial chromosomes not built linearly and operations change order, this will be complicated: metric value on genes instead of observations?
								HCAs: Since unlike GAs, we only have one solution candidate at any time, the metric value will only need to be changed in that single solution.
								Spike: I do not see any support for this.
								GM: It is easy to update the score of observations based on what was just scheduled.
Ability to fix and respect band, updating score before running.	4	0	-2	-2	-2	-2	4	ILPs: Since this is a black box, changing the metric when something has been scheduled is not really an option. The problem is fixed and handed off to the ILP scheduler.
(i.e. before scheduling, calculate new scores as metric changes)								GAs: This may possible, but difficult because due to the constant shifting nature of the schedule, nothing is ever truly scheduled until the chromosome is fully generated.

Pugh Matrix for Scheduler Trade Study								
Glossary:	**Weight**	The relative importance of the requirement (from 0, i.e. least important, to 5, i.e the most important)						
	QC	Queue Coordinator						
	ILPs	Integer Linear Programs (Gurobi, which possible Multi-Objective Function Extension when stated)						
	GAs	Genetic Algorithms (with possible Multi-Objective Evolutionary Algorithm Extension when stated)						
	HCAs	Hill-Climbing based algorithms						
	Spike	Spike algorithm for Hubble Scheduling						
	GrMax	Bryan Miller's Greedy-Max algorithm						
Requirements	**Weight**	**QCs**	**ILPs**	**GAs**	**HCAs**	**Spike**	**GrMax**	**Notes**
								HCAs: Suffers from the same problem as GAs.
								Spike: Unknown, but suspected not to be possible.
								GM: Possible to do.
Timing constraints	5	0	1	0	1	2	1	ILPs: Timing constraints can be represented in the model by modifying or adding additional constraints. If observation o cannot be scheduled at time slot t, set y_ot to 0 or omit it completely from the constraints.
								GAs: Timing constraints can be established in initial population but we must store them and make sure operations do not result in chromosomes that violate them.
								HCAs: Unlike GAs, having only one solution, it would be easier to make sure that the neighborhood does not contain candidates that violate timing constraints.
								Spike: Spike handles timing constraints automatically.
								GM: Supported, part of visibilty calculations
Flexible sequence length - stop an observation at an atom	5	0	0	0	0	0	0	The stopping should be the responsibility of the seqexec, and the algorithm should be rerun if necessary to get a new schedule.
Flexible sequence length - use atoms to build up sequences	4	0	-2	-1	-1	-2	1	ILPs: Atoms slow the problem down specifically as additional constraints are needed to ensure they are executed in the correct order, and building up atoms dynamically does not seem possible.
								GAs: Given the heuristics, building up atoms could be challenging: making operators respect atoms and not tear apart built up atoms would be very difficult.
								HCAs: This depends on how a neighborhood is defined, but suffers from some of the same problems as GAs, with perhaps less difficulty due to less dramatic changes to the schedule.
								Spike: Seems unable to handle atoms and as a black box, building them up is unlikely to be possible without code modification.
								GM: Will schedule as much of an observation is possible, building up atoms.
Supports AND logic	5	0	0	-2	-1	-2	0	ILPs: Trivially easy to do by adding a constraint for each AND as per the Las Cumbres paper.
								GAs: Possible, but more difficult, as chromosomes without the necessary genes will have to be filtered, and finding chromosomes with all genes might be a challenge.
								HCAs: SImpler than GAs, as consider only neighborhood to be members that meet the neceeesary criteria.
								Spike: I do not believe that this is supported by Spike.
								GM: Should be possible with some extra logic or by using timing constaints, needs prototyping
Supports OR logic	5	0	1	-1	0	-2	0	ILPs: For XOR, can only be done for (at least) two observations. For standard logic OR, should be able to be done across arbitrary numbers of observations by adding a constraing for each OR. (O1 OR O2 OR O3 as O1 + O2 + O3 >= 1)
								GAs: ORing observations poses some similar problems as with AND. In the case of general OR, this should not be too difficult. In the case of XOR, this will be very difficult, particularly in long AND - OR expressions due to small probabilitiy of rule being met.
								HCAs: Similar to the GA problem, although since there is only a single solution, we can maintain it at every ppoint.
								Spike: I do not believe that this is supported by Spike.
								GM: Should be possible with some extra logic or by using timing constaints, needs prototyping
Calculate a score for a time period	5	0	1	1	1	1	1	All automated schedulers compute comparable scores. There is currently no implemented QC scoring, but one is being developed.
Make plan for (at least) an entire night	5	0	0	0	0	0	0	All: trivial.
Appropriate for short-term scheduler	1	0	2	2	2	2	2	ILPs: Can be used as real time, medium, and long-term schedulers.
								GAs: Can be used as real time, medium, and long-term schedulers.
								HCAs: Can only be used as real time / nightly schedulers.
								Spike: Can be used as real time, medium, and long-term schedulers.
								GM: Designed mainly to schedule a single night
Appropriate for medium-term scheduler	1	0	2	2	-2	2	-1	ILPs: Can it be used to determine when gratings/masks should be installed/removed?
								GAs: Can it be used to determine when gratings/masks should be installed/removed?
								HCAs: Inapplicable.
								Spike: Can it be used to determine when gratings/masks should be installed/removed?
								GM: Could be used in future simulation mode with multiple iterations, multi-night single-pass mode needs prototyping. maybe not the most appropriate method
Appropriate for long-term scheduler	1	0	2	2	-2	2	-1	ILPs: Can it be used to determine when instruments should be mounted or classical runs scheduled?

Pugh Matrix for Scheduler Trade Study								
Glossary:	**Weight**	The relative importance of the requirement (from 0, i.e. least important, to 5, i.e the most important)						
	QC	Queue Coordinator						
	ILPs	Integer Linear Programs (Gurobi, which possible Multi-Objective Function Extension when stated)						
	GAs	Genetic Algorithms (with possible Multi-Objective Evolutionary Algorithm Extension when stated)						
	HCAs	Hill-Climbing based algorithms						
	Spike	Spike algorithm for Hubble Scheduling						
	GrMax	Bryan Miller's Greedy-Max algorithm						
Requirements	**Weight**	**QCs**	**ILPs**	**GAs**	**HCAs**	**Spike**	**GrMax**	**Notes**
								GAs: Can it be used to determine when instruments should be mounted or classical runs scheduled?
								HCAs: Inapplicable.
								Spike: Can it be used to determine when instruments should be mounted or classical runs scheduled?
								GM: Same as above.
Short-term scheduler performance	5	0	1	2	2	2	2	ILPs: Can meet performance requirements with Gurobi. Unsure about the open source schedulers. Time can be improved by accepting less optimal solutions, but cannot specify time.
								GAs: Can be made multithreaded and we can stop algorithm after a certain amount of time. (More time = better results.)
								HCAs: Same as GAs.
								Spike: More timing calculations have to be done with larger data sets, but on the tested data sets, the requirement is met.
								GM: <10sec to make a single-night plan for both telescopes
Medium-term scheduler performance	4	0	1	2	-2	2	0	ILPs: This depends on the ILP solver, the number of variables and constraints, and the amount of data. Further testing is needed.
								GAs: Again, GA runtime can be controlled, although without sufficient time, the solutions might be reasonably far from optimal.
								HCAs: Inapplicable. HCAs ddo not constitute a medium-term scheduler.
								Spike: Unknown. More tests will have to be done; however, it seems feasible that the bound could be met.
								GM: This has not been benchmarked recently, if run serially then the performance is ~ 10sec x N nights, so it would meet the requirements.
Long-term scheduler performance	3	0	1	2	-2	2	0	ILPs: Same as above .
								GA: Same as above.
								HCAs: Same as above.
								Spike: Same as above.
								GM: Same as above.
Quality of results	4	0	1	0	0	0	0	ILPs: We can guarantee results of a certain quality, although this affects runtime. Tends to produce better results that GAs.
								GAs: We cannot guarantee results of any quality regardless of runtime due to stochastic processes and must still compare to QC scoring.
								HCAs: We cannot guarantee results of any quality regardless of runtime due to stochastic processes and must still compare to QC scoring.
								Spike: We cannot guarantee results of any quality due to differing algorithms in pacakage.
								GM: Adjust scoring and algorithm to get reasonable results, cannot guarantee that this is the most optimal
Testable (run future simulations)	5	0	1	1	1	1	1	In all cases, simply provide data for future and run the algorithm to receive results.
Testable (run on past data)	5	0	1	1	1	1	1	In all cases, it should be possible to execute the scheduling at any point in the past.
Schedules calibrations (night and day)	5	0	-1	-2	-2	-2	0	ILPs: Nighttime calibrations would have to be specified as inputs and we would need to know how close to their observations they need to be.
								GAs: Same as ILPs, and chromosomes would have to be verified to each make sure this closeness was maintained. Danger of calibrations being separated from their observations through operations must be addressed.
								HCAs: Similar dangers to GAs: initial random candidate must reflect this,and neighborhoods should be pruned to maintain the characteristic.
								Spike: Unsure if this is possible.
Can optimize calibrations (shared)	3	0	-2	-2	-2	-2	-1	ILPs: We would have to do this manually or by coming up with a separate model to pre-optimize the calibrations before running the scheduler optimization.
								GAs: We would have to do this manually.
								HCAs: We would have to do this manually.
								Spike: Unsure if this is possible.
								GM: Maybe possible with extra logic, needs prototyping
Reducing unscheduled time	3	0	1	2	2	2	0	ILPs: The ILP solver reduces unscheduled time naturally with few exceptions, e.g. if a schedule exists with a very high score that has a hole where nothing can fit. A multi-objective function could be used to minimze unscheduled time with Gurobi.
								GAs: We can reduce unscheduled time by inserting observations, shuffling observations to conglomerate gaps (at a possible expense of fitness), and then inserting observations into the gaps. Could also use MOEA.

Pugh Matrix for Scheduler Trade Study

Glossary:	**Weight**	The relative importance of the requirement (from 0, i.e. least important, to 5, i.e the most important)
	QC	Queue Coordinator
	ILPs	Integer Linear Programs (Gurobi, which possible Multi-Objective Function Extension when stated)
	GAs	Genetic Algorithms (with possible Multi-Objective Evolutionary Algorithm Extension when stated)
	HCAs	Hill-Climbing based algorithms
	Spike	Spike algorithm for Hubble Scheduling
	GrMax	Bryan Miller's Greedy-Max algorithm

Requirements	**Weight**	**QCs**	**ILPs**	**GAs**	**HCAs**	**Spike**	**GrMax**	**Notes**
								HCAs: We can reduce unscheduled time by careful neighborhood design.
								Spike: Spike should automatically reduce unscheduled time based on its algorithms.
								GM: Schedules as much of a night as possible, splitting observations helps.
Minimizing acquisition and slew times	4	0	-2	-2	-2	-1	-1	ILPs: Same as above.
								GAs: Same as above.
								HCAs: Same as above.
								Spike: I am unsure if Spike offers this functionality. More experimentation would be needed.
								GM: Maybe possible with extra logic, needs prototyping
Maximizing semester metric	5	0	2	1	0	-1	-1	ILPs: We can guarantee results of a certain quality, although this affects runtime. Tends to produce better results that GAs.
								GAs: We cannot guarantee results of any quality regardless of runtime due to stochastic process.
								HCAs: As with GAs, we cannot guarantee results of any quality regardless of runtime due to stochastic processes; however, GAs tend to yield better results that simple heuristics.
								Spike: Spike offers five algorithms to trade off between various features. Many of these ignore priority. More experimentation is needed but the initial results do not look promising.
								GM: It is trying to maximizing a score that includes the metric but the result may not be optimal
Prioritizing already started observations / programs	4	0	2	2	2	-2	2	ILPs: This should be done by increasing the metric for those observations / programs.
								GAs: This should be done by increasing the metric for those observations / programs.
								HCAs: This should be done by increasing the metric for those observations / programs
								Spike: Spike only takes priority into account in one of its algorithms, so priority cannot be used outside of this algorithm to prioritize.
								GM: Handled easily since the metric of remaining observations can be updated with each scheduling iteration
Preferentially observe at low air mass	3	0	0	0	0	2	0	ILPs: This can be encoded in the objective function, with hour angle or airmess term included in score.
								GAs: This can be encoded in the fitness function, with hour angle or airmess term included in score.
								HCAs: This can be encoded in the fitness function, with hour angle or airmess term included in score.
								Spike: I believe that Spike automatically tries to do this already.
								GM: Hour angle or airmass term included in score
Bump for thesis / DD programs	3	0	0	0	0	-1	0	ILPs: Increase metric for thesis / DD by visibility calculations (limited program length) or by a metric boost
								GAs: Increase metric for thesis / DD by visibility calculations (limited program length) or by a metric boost
								HCAs: Increase metric for thesis / DD by visibility calculations (limited program length) or by a metric boost
								Spike: Unsure of how to do this as only one algorithm takes priority into account. Could possibly work with timing windows to drive higher?
								GM: Increase metric for thesis / DD by visibility calculations (limited program length) or by a metric boost.
Avoidance timing window (no guide star)	4	0	1	-1	-1	-1	1	ILPs: Precalculate this information at the beginning of the semester and then modify constraints to disallow observations to be scheduled at these time slots.
(A timing window issue)								GAs: Same as ILPs. Chromosome operations again may result in invalid chromosomes and they will have to be checked.
								HCAs: Same as ILPs. Neighborhoods must be maintained to prune invalid solutions.
								Spike: I believe that this information would have to be sent to Spike in the form of timing windows?
								GM: Pre-calculate then handle with timing constraint
IF chains: If A is done, then B cannot be done for X hours	3	0	2	0	0	0	0	ILPs: As a timing constraint: for each, add t constraints that force observations to be done in a specified order with arbitrary time gaps between them. (e.g. O, and then P not for 3 hrs: for all t: Ot - P0 - P1 - .... - Pt - P(t-1) - P(t-2) = 0)
(weakness of the current system)								GAs: To achieve this adds substantial difficulties in both the initial population creation, as well as the operations: in building initial population, would have to generate candidates until they satisfy this, and discard candidates that do this after GA ops are done.
								HCAs: Similar problems as with GAs, but on a single scale solution and movements of that solution through the space.
								Spike: I don't believe that this functionality is supported.
								GM: Should be possible with extra logic or timing constraints, needs prototyping
Adjust priority of particular program for fraction of night	2	0	2	2	2	0	1	ILPs: Simply adjust score for program during the relevant time slots when building objective function? Information could be stored in resource table.

Pugh Matrix for Scheduler Trade Study

Glossary:	**Weight**	The relative importance of the requirement (from 0, i.e. least important, to 5, i.e the most important)
	QC	Queue Coordinator
	ILPs	Integer Linear Programs (Gurobi, which possible Multi-Objective Function Extension when stated)
	GAs	Genetic Algorithms (with possible Multi-Objective Evolutionary Algorithm Extension when stated)
	HCAs	Hill-Climbing based algorithms
	Spike	Spike algorithm for Hubble Scheduling
	GrMax	Bryan Miller's Greedy-Max algorithm

Requirements	**Weight**	**QCs**	**ILPs**	**GAs**	**HCAs**	**Spike**	**GrMax**	**Notes**
(time score increases): e.g. we have a visitor for frac of night								GAs: Similar to ILPs, adjust score for program during that part of night when building fitness function?
(N.B. Either not done in the current system or done manually.)								HCA: Similar to ILPs, adjust score for program during that part of night when building fitnress function.
								Spike: Again, due to the lack of priority in most calculations, this may be dificult to do. May be able to exploit available resources to accomplish this.
								GM: Handle with timing constraints or adjust score function
Fix observations into a scheduling period (O0 < O < O1?)	3	0	0	0	-1	-2	0	ILPs: This is similar to IF chains, but more complex. Could be formulated as timing constraints, whicn wouldn't require a large increase in prroblem size.
								GAs: This will require careful creation of initial population and checking the chromosomes in each generation to ensure that validity is maintained, or use timing constraints.
								HCAs: The initial candidate would have to have this quality and each subsequent neighbouhrhood would have to be composed strictly of solutions that maintain it.
								Spike: I am unsure if Spike offers this functionality. I suspect not.
								GM: Maybe possible with extra logic or timing constraints, needs prototyping
Internal observation priorities	3	0	-1	-1	-1	-2	0	ILP: Should be factored into the metric formula? Needs discussion and investigation, if factored into metric/score then this is no longer an internal priority.
								GA: Factor into metric or add logic. Needs prototyping.
								HCAs: Factor into metric or add logic. Needs prototyping.
								Spike: I suspect Spike cannot do this.
								GM: Factor into metric or add logic. Needs prototyping.
Score:		0	62	29	1	-21	91	