**COMP5920M Schedulig**

**Q1:**

**(a): (i):**

As for long term planning for UK rail transport, it’s usually generate regular operations timetables that can last for a very long period, like several months or half a year. Thus, optimisation is very important.

Also, for the train planning process like taking the unit diagrams as main input. It require the train unit diagrams are finalised, so that it can move on to the crew scheduling. If the train unit diagram is not good enough, it will effect the crew scheduling. So for long term planning, people need to optimize the schedule as good as possible. For example, minimise the crew cost for crew scheduling, and minimise operating cost for train unit scheduling.

The alogrithm or apporach that long term planning usually used are GaS, ILP, etc. These requires much more time and effort to generate, calculate and select a good schedule.

As for very short term planning, it’s usually used for re-generate a schedule or solution for unplanned disruptions. For example, if a train crew is calling a sick or a train unit is breakdown, it require some real time information for subtitution and re-scheduling. Thus, it also require a very short term planning to generate the a solution or timetable.

People tend to use heuristic for very short time planning. It’s may not good or optimisation enough. But it works, can process very fast and it can be done in a short period.

**(ii):**

Planning horizons are aim to give suitable solution or timetable to specific problem.

The real world problem like franchise or council tender bids for many years, may perfer approach like ball park estimates, which require no detailed operational schedules.

Some of the real world problem like regular operations for half-yearly timetables, is predicatable and can be used for a long time once the timetable is generated. The approach is focus more on the degree of optimisation and reduce cost. It’s tend to find the global optimisation rather than the local one. So approach like generate and select approach can be useful in this problem.

Some of the problem like holiday or maintenance timetable, is predictable service changes and need re-scheduling in advance. The best way to do is to minimize the original timetable changes and improve the schedule robustness against unexpected, such as delays.

Other problem like unplanned disruptions. It’s allow the solution may not good enough or only reach local optimisation, but generate fast. Approach like heuristic are prefect match it’s requires. And people would use such approach to deal with short term problems.

**(b):**

* Bus fleet size estimation. Find the target number of buses in service, which means the number of simultaneous trips.
* Construct an initial solution. Assign number of buses to the earliest departures. Then link each assigned trip to it’s earliest unassigned and feasible successor repeatly. Then assign any trips remaining to the first bus.
* Iterative improvements. To minimise the infeasibility, deadrunning time and idle time.

Pairing up solution components and restructuring them in a simple two-way swap.

Design a acceptance criteria. Compare the cost and penalty of the old and new solution. The solution can be link based or block based.

* Then modelling a swap operation. First, define contents to be swapped, like one trip and links before and after on the same bus. Then define how the swap contents are updated, like trips at the end of the links are swapped.

**(c):**

It require the knowledge from many fields, like computer vision, scheduling , distributed system, cloud computing, edge computing, psychology and so forth.

* For computer vision. To be able to use the suitable technologies to do the image segmentation like the pedestrain recognition and motion detection like tracking. The technological compromise would be more clear.
* About the distributed system and edge/cloud computing. These technology can be useful in this case. Each camera can be regard as a network unit, which can upload data or video stream to the cloud, and process the video, analyse in the cloud or local to get the result that people may wanted. Leader should test if this would work and feasible for this system.
* Psychology also important. Leader should come up with ideas to tell the difference between normal and strange behaviour. For example, theft always look around and act sneakily. Using technology to distinguish behaviour according to these characteristic.
* Scheduling is the most important. It’s like the central role integrating different field of knowledge into a solution menthod.

The guys with these knowledge in different fields has much more benefit of leading the project. He/ she would be able to consider in the contextes of other discipline and can have a good understanding of the best expertise in all the disciplines involved.

**Q2：**

**(a): (i):**

A timetable should be given to form a set of train nodes. Each nodes represent a trip along with it’s time, seats demand, arr/dep station and so on.

Then add source and sink nodes to the graph.

Assess and insert potential arcs, and rule out many potential arcs due to time infeasible, etc.

Other possible arcs, like several kinds of time allowances may be needed. Empty running train nodes may be inserted.

Finally, arcs from/to source/link inserted.

**(ii):**

This model not consider station infrastructure. So in the optimal solution that the model generated may exist unit blockage.

This model not include shunting operation. So the solution may lack of platforms, siding and depots assignment as well as the paths of shunting movements.

Finally, this model also not include the unit position in a coupled formation. Like the unit poistion is in the front or middle or rear of the train. It’s also lead the solution interrelated with blockage and shunting.

**(iii):**

Train units can coupling or decoupling according to the different seats demand. As for it’s fleet size, which is used in vampires algorithm, it would be dynamic changed and more complex. It’s not like the bus problem, which the fleet size can be estimated easily and improve.

Train unit scheduling can have more constrain than bus scheduling. For example, the train have to run on the railway, then shunting movements may be needed. And for vampires algorithm, it’s not considering that.

Train also may have potential blockages, it won’t be included in vampire algorithm. In addition, in swap operation, it may cause unnecessary blockages.

Train units sometimes won’t fit in some platform, due to it’s length. And this alogorithm also cannot add this condition as constrains.

Train unit would take more time to turn around. And some platforms are through platform, some are deadend platform. This can be infeasible and not being considered in the vampire algorithm.

**(b):**

Generate and select approach (GaS) can be useful in crew scheduling. Similar to SLIM, they both use integer linear programming (ILP) as powerful solver. ILP are used in select phase, and it’s the most successful based on set covering model at present. ILP aim to select few shift from the enormous pot of candidates, and it’s focus more on optimising if an optimal solution exists.

And branch & bound (B & B) also can be useful for the solution of integer. Both SLIM and GaS would use B&B technique. Specialised branching strategies and divides the search space more efficiently. Progressively adds extra constraints can be considered at B&B. B&B are really helpful to solve large problems.

As for the initial instance or solution can be low quality in both SLIM and GaS. Heuristics are both used in SLIM and generate pahse. It’s suitable and fast.

Finally, iterative convergence to a high quality solution can be generated for both SLIM and GaS

**Q3:**

1. **(i):**

First, in the generate phase, we can generate huge number of staff shifts without considering of optimization or if a shift would fit well with others. But all of the shifts should be legal and operable. Use heuristic reduction to make the number of shifts be smaller, like reduce relief opportunities. Focus on several charactistics like work to be covered by staff, some hard rules, soft rules and some preferences.

Then in the select phase. Use the enormous pot of candidates shifts from generate phase as input. And use ILP to optimize the solution, select the minimise cost solution. Set covering model, set ground elements like work piece in flight diagram, and staff diagram as subset. Set covering crew scheduling ILP, clearlify the objective function to minimise the cost and number of cabin staff. Some side constraint may needed. And set lower/upper bound. Then use the revised simplex method to solve the relaxed LP. Column generation also useful for solving the LP. Then use the branch and bound technique to derive an all integer solution. With the help of heuristics can speed up the computation.

**(ii):**

The difficulity of two problem using the same approach is determined by their constraints. The airline would be easier. As for airline crew, they don’t need relief opportunity during the work day. They can finish their lunch or break during the flight. Espically for long-distance non-stop flight. The cabin staff can work and relief in the work time. And their shifts begin and end in the same city. Flight crew may not need extra travel, which reduce the problems complexity.

**(iii):**

We can use this rule to reduce the pot of candidate shifts before it as input to the ILP.

Repeat

If ( > ):

Remove j from 

During the ILP, the cost  is positive correlation with the length for each shifts.

**(b):**

The idea is great, it would get closer to global optimality in train operations planning, only if it’s implemented. The algorithm has to deal with the constraints of crew scheduling as well as the unit scheduling. And the parameter for crew and train unit also have to be considered at the same time. Massive data for train unit and crew will increase the calculated amount and the complexity.

**(c):**

In bus vehicle scheduling, the shortest paths can improve the matrix for bus route. It can find the short path between locations and save time and cost. Some times the deadruns are quicker, it’s also one way to save time. So in this way, we try every cell as a potential via location, and test which martix cell could benefit using it. After each cell value is updated, previous searches have to be repeated. So many iterations can increase the time complexity of the computation. But the timetable after floyd’s algorithm is much more efficient and minimise cost compare with the previous timetables.

For train crew scheduling, the extra travel for crew shift can be regrad as wasting of time just like the deadrun. But sometimes it can reduce the cost. For example, let some of the crew travel to a large station with many early trains need to be operated. And compute every possbilities point for possible improvements, and find the most benefit one then update the value. Then repeat the precedure until no value is updated.

It would be more straightforward when the shortest path applied into bus scheduling. Just find the shortest path or faster path then update the bus route martix. And it would be abstract when apply to crew scheduling. But it also means try all the possible improvement to improve the schedule. The computation for time complexity and space complexity are both large for both bus and crew scheduling, due to vast iterations and recursion.