Current implementation doesn’t account for overlap in the implementation of the restricted section.

Current implementation has a hack to ensure that we use all sub event. Remember we were always getting a max of 54 events instead of 55. This was because the number of available sub events workable with the last free space was always less than the number of available sub events. We used a hack to force all available subevents to be available for assignment

Currentnt Location initialization does not take into account descrition initialization

There is no current way to update a subevents location. “Update Subevent” doesn’t update the location”

Current implementation does not place Subcalevent in the most optimized location in PlaceSubCalEventInLowestCostPosition

Current implementation has no implementation for shortest path for subcalevent in getArrangementWithLowestDistanceCost

There is a bug with snug array, it doesn’t seem to work correctly with the log file, PseudoScheduleCookedValues.xml. It generates only one snug possibility in the first free timeline, however, after your calculation, we should have at least 8viable solutions. Check the image on skydrive, name of file WP\_20140122\_003.jpg

There is a potential bug pertaining to how you handle partials. The current implementation of partial for a timeLine works by setting a limiting timeframe between the variable “EarliestReferenceTIme” and the first restricted time space or First partial, depending on whichever comes first. The after the code executes the best fit using the available SubCalendar event, it updates the EarliestReferenceTIme to the end of this time frame. This poses a risk in which one might have several partials with deferring CalendarEvent start times that are relatively closed to each other. If they are too close, a best fit wont work because the subcalendar events might be too small to fit between the time frame. Look at image below

Start 10:00A A-12:30p B-01:00pm Restricted 1:20pm

| | | |

Assuming the space between Start(10:00A) and Event “A” 12:30p is occupied. The reference start time will get updated to 12:30p since that was the end time from the preceeding calculation. The new end time will be 1:00p for Event “B” since it is befire the restricted Event. Assuming we are trying to schedule an event “C” of 40 mins it wont gain access because of the timespan between EarliestReferenceTime, Event “A” start time, and Event “B” calendarevent partial start time. This goes on to update the EarliestReferenceTime as Event B’s which is 1:00. The new time frame for the next free spot will be 1:00pm to the restricted start time which is not 1:20pm. In an ideal situation, the Event C should be able to insert anywhere as long as it’s a free space like 12:30p to 1:20pm

You need to write code that enforces the highest combination of subcalevents with deferring IDs