A Computer Implemented Method and Apparatus for obtaining an Optimal Project Makespan/Cost

Background and Field

The invention relates to a computer implemented method and apparatus for obtaining an optimal project makespan and/or cost.

Many real-life situations require simultaneous management of multiple projects, some of which may be required to share a common pool of resources. For example, a vessel docked at a shipyard often requires multiple tasks or projects to be performed concurrently on it, with an overall objective of having the vessel ready for use within a predetermined period of time. In another example, two or more vessels may need to be maintained using a common pool of resources such as welding personnel, equipment and tools and it can be a challenged to manage the maintenance projects on the different vessels using the common pool of resources, especially under time constraints. Other examples may be observed during aircraft maintenance where overhaul, repair, inspection and/or modification of an aircraft must be completed before a specified time to avoid costly penalty charges, and where multiple tasks or projects typically run concurrently. Yet another example where the management of multiple projects may share common resources is in construction or building situation. Typically, a manager of extensive experience would be required to allocate the limited resources among the multiple projects such that all the projects can be completed within a prescribed time. Even so, the manager has no systematic

means of determining ahead of time the optimal way of allocating the limited resources, and is pressed to provide cost and makespan estimates based on experience and gut-feel. It is also impossible for the manager to compare across different schemes of resource allocation.

It is an object of the present invention to provide a computer implemented method and apparatus for obtaining an optimal project makespan and/or cost which alleviates at least one of the disadvantages of the prior art and/or to provide the public with a useful choice.

Summary

In a first aspect, there is provided a computer implemented method of obtaining an optimal project makespan, comprising (i) receiving a plurality of tasks for completing the at least one project within an initial project makespan, identifying at least one critical task from the plurality of tasks which determines the initial project makespan, deriving a critical chain for the at least one project based on the at least one critical task; (ii) allocating additional resource to the at least one critical task to obtain a minimum project makespan of the at least one project; (iii) deriving a current project cost based on the minimum makespan; (iv) if the current project cost is equal or less than a predetermined budget, identifying the minimum makespan as the optimal project makespan; (v) if the current project cost is more than the predetermined budget, adjusting the allocation of the additional resource to the at least one task to derive a project cost which is equal or less than the budget; the project cost being associated with a revised

project makespan; and (vi) identifying the revised project makespan as the optimal project makespan for the at least one project.

The method may comprise calculating a minimum project cost for the at least one project; and comparing the minimum project cost with the budget. In this way, it is possible to determine if the budget is realistic or not. If the minimum project cost is more than the budget, the method further comprises outputting that the budget is not able to complete the at least one project.

There may be more than one critical task assigned with the additional resource, and step (ii) includes tracking change in cost after allocation of the additional resource to the critical tasks. This is particularly useful to track quantitative changes which may be used for further adjustments later. The method may further comprise summing up all the change in cost to derive a total change in cost; and deducting the total change in cost from the current project cost to obtain the minimum project cost.

If the current project cost is more than the budget, step (v) of the first aspect may include un-allocating the allocation of the additional resource to derive the project cost which is equal or less than the budget. In other words, the un-allocation may be repeated until the project cost comes to within the budget. The un-allocation may include arranging the critical tasks in order of the tracked change in cost; each critical task being associated with a said additional resource; and un-allocating the said additional resource one at a time to derive the project cost which is equal or less than the budget.

The method may be applied to multiple projects and thus, step (i) may include deriving a critical chain for multiple projects, and the method may include obtaining the optimal project makespan for each of the multiple projects or for all the multiple projects.

In a second aspect, there is provided a computer implemented method of obtaining an optimal project cost, comprising (i) receiving a plurality of tasks for completing the at least one project within an initial project makespan, identifying at least one critical task from the plurality of tasks which determines the initial project makespan, deriving a critical chain for the at least one project based on the at least one critical task; (ii) allocating additional resource to the at least one critical task to obtain a minimum project makespan of the at least one project; (iii) calculating a minimum project cost based on the allocation of the additional resource; (iv) deriving a current project makespan for the project based on the minimum project cost; (v) if the current project makespan is equal to or less than a predetermined duration limit, identifying the minimum project cost as the optimal project cost; (vi) if the current project makespan is more than the predetermined duration limit, adjusting the allocation of the additional resource to the at least one task to derive a project makespan which is equal or less than the predetermined duration limit; the project makespan being associated with a revised project cost; and (vii) identifying the revised project cost as the optimal project cost for the at least one project.

Step (ii) of the second aspect may comprise comparing the minimum project makespan with the duration limit. In this way, it is possible to determine whether the duration limit is realistic or not. If the minimum project makespan is more than the duration limit, the method may further comprise outputting that the duration limit is not able to complete the at least one project.

There may be more than one critical task assigned with the additional resource, and step (iii) of the second aspect may include tracking change in cost after allocation of the additional resource to the critical tasks. This is useful to track quantitative change in the cost attributed to the allocation of the additional resource. The method may further comprise summing up all the change in cost to derive a total change in cost; and deducting the total change in cost from the current project cost to obtain the minimum project cost.

Step (vi) of the second aspect may include un-allocating the allocation of the additional resource to derive the project makespan which is equal or less than the duration limit. The un-allocation may include arranging the critical tasks in order of the tracked change in duration; each critical task being associated with a said additional resource; and un-allocating the said additional resource one at a time to derive the project makespan which is equal or less than the duration limit.

Just like the first aspect, the second aspect may be applied to multiple projects, and step (i) may include deriving a critical chain for multiple projects, and the

method may include obtaining the optimal project cost for each of the multiple projects or for all the multiple projects.

The additional resource in the above aspects/features may include overtime and/or outsourcing of tasks to subcontractors. If the additional resource allocated to the at least one critical task includes outsourcing of tasks to subcontractors, the outsourcing of the tasks may replace an initial resource allocated to the at least one critical task under step (i).

Step (i) of either or both aspects may include assigning constrained resources shared with other projects to the plurality of task, and leveling the constrained resources. The method may also comprise, prior to step (i), receiving user definition of the plurality of task required for completing the at least one project and initial resources for allocation to each of the task for deriving the critical chain. Step (i) may also include a preprocessing step to adapt the method to be compatible with a third party project management application. The above methods may also comprise receiving task priorities, and associating the task priorities with each one of the plurality of task.

During the calculation of the minimum project cost in either aspects, the plurality of tasks may include non-critical tasks, at least some of which are associated with additional resources; and the method may comprise tracking change in cost after allocation of the additional resources to the non-critical tasks, and adding the tracked change in cost of the non-critical tasks to the total change in cost for the calculation of the minimum project cost.

Optionally, step (i) in either aspect may include (a) identifying a set of critical tasks for the or new critical chain, at least some of which are associated with additional resources; and step (ii) may include (b) checking if any of the critical task's duration has not been adjusted due to the associated additional resource, and if there is, identifying such critical tasks; (c) adjusting the duration of such critical tasks based on the associated additional resource; (d) deriving a new critical chain based on the adjusted duration; and (e) repeating steps (a) to (d) until the duration of all the critical tasks associated with the additional resource have been adjusted to derive a revised critical chain, and obtaining the minimum makespan based on the revised critical chain. As a result, it can be appreciated that the critical chain is dynamic and may change depending on the additional resource allocated to the tasks.

The method discussed above may take several forms. For example, the method may be implemented as a plug-in computer program product which forms a third aspect and the product comprises instructions when executed are arranged to control a computer to perform the method as discussed above.

There may also be a computer program product which forms a fourth aspect and this is stored on a data recording medium comprising instructions when executed are arranged to control a computer to perform the method as discussed above.

An embodiment of the invention may also be implemented in hardware, which forms a fifth aspect, and this is in the form of apparatus for obtaining an optimal project makespan. The apparatus comprises a processor configured to (i) receive a plurality of tasks for completing the at least one project within an initial project makespan, identify at least one critical task from the plurality of tasks which determines the initial project makespan, derive a critical chain for the at least one project based on the at least one critical task; (ii) allocate additional resource to the at least one critical task to obtain a minimum project makespan of the at least one project; (iii) derive a current project cost based on the minimum makespan; (iv) if the current project cost is equal or less than a predetermined budget, identify the minimum makespan as the optimal project makespan; (v) if the current project cost is more than the predetermined budget, adjust the allocation of the additional resource to the at least one task to derive a project cost which is equal or less than the budget; the project cost being associated with a revised project makespan; and (vi) identify the revised project makespan as the optimal project makespan for the at least one project.

In a sixth aspect, there is provided apparatus for obtaining an optimal project cost, comprising a processor configured to (i) receive a plurality of tasks for completing the at least one project within an initial project makespan, identify at least one critical task from the plurality of tasks which determines the initial project makespan, derive a critical chain for the at least one project based on the at least one critical task; (ii) allocate additional resource to the at least one critical task to obtain a minimum project makespan of the at least one project; (iii) calculate a minimum project cost based on the allocation of the additional

resource; (iv) derive a current project makespan for the project based on the minimum project cost; (v) if the current project makespan is equal to or less than a predetermined duration limit, identify the minimum project cost as the optimal project cost; (vi) if the current project makespan is more than the predetermined duration limit, adjust the allocation of the additional resource to the at least one critical task to derive a project makespan which is equal or less than the predetermined duration limit; the project makespan being associated with a revised project cost; and (vii) identify the revised project cost as the optimal project cost for the at least one project.

Other embodiments of the present invention may provide methods of managing a portfolio of projects, each portfolio having at least one project. Each project may further be divided into one or more tasks. The management of the portfolio may be performed with an objective of a desired outcome, in which the desired outcome may be to minimize the makespan of the projects or to minimize the cost incurred by the projects. The projects and/or tasks may run concurrently, with different or same start times. The makespan of the portfolio refers to the period beginning with the earliest start time and ending with the latest end time of all the projects in the portfolio. The management of the portfolio may take into consideration resources in the form of overtime and outsourcing. Outsourcing may include engaging temporary staff or independent contractors (also known as sub-contractors) to perform specified projects or specified tasks. Overtime refers to having staff perform specified projects or specified tasks, over and above their usual working hours, and at an overtime wage rate typically higher than the normal wage rate. The management of the portfolio may provide

methods related to critical chain methodology to schedule the projects or tasks, or to allocate resources to the projects or tasks. Two or more projects or tasks may involve one or more similar or identical resources, such that the resources have to be "shared" between the projects or tasks in a portfolio. Each of the resources may be a "non-consumable" resource in the sense that the resource may be used by a first project or first task at one time, and the same resource may be used by a second project or second task at another time.

Other embodiments of the present invention in which the desired outcome is the shortest makespan may include methods having steps of: i) identifying or finding a critical chain for a portfolio of projects, in which each project may be expressed in terms of at least one task, thereby identifying one or more critical task; ii) pre-processing the projects to be optimized; iii) assigning overtime and subcontractor to each critical task if so desired; iv) finding a minimal project makespan and a minimal project cost based on heuristic rules that adjust the duration and/or cost of each task; v) identifying any task which is not assigned to overtime or outsourcing so that the cost of each project may be kept within a pre-determined budget; vi) finding an optimal makespan for the portfolio of projects based on a list sorted by changes in task durations; and vii) post-processing.

Optionally, embodiments of the present invention may further include, when identifying or finding a critical chain, leveling the resources for a plurality of projects, and normalizing the resources so as to resolve resource conflicts between the plurality of projects.

Other embodiments of the present invention in which the desired outcome is the lowest cost may include methods having steps of: i) identifying or finding a critical chain for a portfolio of projects, in which each project may be expressed in terms of at least one task, thereby identifying one or more critical task; ii) pre-processing the projects to be optimized; iii) assigning overtime and subcontractor to each critical task if so desired; iv) finding minimal project cost and minimal project makespan based on heuristic rules that adjust the duration and/cost of each task; v) identifying any task which is not assigned overtime or outsourcing, so that the project makespan would be kept within a pre-determined time; vi) finding optimal cost for the portfolio based on a list sorted by incremental costs; and vii) post-processing.

Optionally, embodiments of the present invention may further include, when identifying or finding a critical chain, leveling the resources for a plurality of projects, and normalizing the resources so as to resolve resource conflicts between the plurality of projects.

In this manner, embodiments of the present invention facilitates systematic scheduling of multiple projects, taking into consideration opportunities for overtime or outsourcing for one or more tasks. There is thus provided a method and software for alternately finding an optimal makespan for a portfolio of projects with constrained project makespan or finding an optimal cost for a portfolio of projects with constrained project makespan.

Brief Description of the Drawings

Examples of the invention will now be described with reference to the accompanying drawings, in which:

Figure 1 is an overview of a system for managing multiple projects which includes an optimization engine;

Figure 2 is block diagram of the optimization engine of Figure 1;

Figure 3 is a flow chart illustrating a method performed by the optimization engine of Figure 2 to obtain an optimal project makespan for a repair and maintenance project;

Figure 4 is a flow chart illustrating a method for un-assigning additional resources in order to derive the optimal project makespan for the repair and maintenance project used as an example in Figure 3;

Figure 5 comprising Figures 5a-5f is a pictorial representation to obtain a minimum project makespan for the repair and maintenance project used as an example in Figure 3;

Figure 6 is an example of a graphical user interface (GUI) for implementing the method of Figure 3;

Figure 7 defines a plurality of task for the repair and maintenance project used as an example in Figure 3 and resources allocated to perform the plurality of task;

Figure 8 depicts a Gantt chart together with the resources of Figure 7 showing a result of optimizing a project makespan for the repair and maintenance project;

Figure 9 is a flow chart illustrating a method performed by the optimization engine of Figure 2 to obtain an optimal project cost for the repair and maintenance project;

Figure 10 is a flow chart illustrating a method for un-assigning additional resources in order to derive an optimal project cost for the repair and maintenance project used as an example in Figure 9;

Figure 11 is an example of a graphical user interface (GUI) for implementing the method of Figure 9;

Figure 12a shows a project schedule for three projects prior to optimizing their project makespans; and

Figure 12b shows the project schedule for the three projects after optimizing their project makespans.

Detailed Description of Preferred Embodiment

Figure 1 is an overview of a system 100 for managing multiple projects according to an embodiment of the invention. The system 100 includes a server 110 and a client 120 which may be a computing device such as a server computer, a personal computer, mobile electronic device, a touch-screen computer or a mobile communications device. The client 120 is communicatively coupled to the server 110 and has an optimization engine 200 configured to communicate with the server 110 and an input/output device 124,

which may include a display monitor, printer, keyboard, touch-screen or mouse etc.

The server 110 includes a project server web services interface 112 arranged to interface the client 120 and an operation information database 114 and a project database 116. The operation information database 114 feeds information relating to production calendar, finance information and resource pool etc to the optimization engine 200. The project database 116 includes resource constraint database 118, project profile database 120 and the project schedule database 122. The resource constraint database 118 provides information relating to all resources including resource constraints such as calendar, critical equipment list etc for completing the tasks in each project. The resources are provided for selection by a user when assigning resource for completion of a task.

The project profile database 120 provides input relating to project information such as profit, due dates, task-resource mapping etc, and the tasks required to complete each project and project schedule database 122 provides scheduling related information such as project tracking information and task block-out information. A project schedule consists of a list of a project's activities/tasks with intended start and finish dates. Project activities/tasks are often estimated in terms of resource requirements, budget and duration, linked by dependencies.

Figure 2 is a block diagram of the optimization engine 200 of Figure 1 which comprises processor 202, an input/output interface 204 and memory 206. The processor 202 performs main processing functions for the optimization engine 200 and communicates with the server 110 via the input/output interface 204. Further, the processor 202 also receives input information from the input/output device 124 and outputs information to the input/output device 124. Information received by or to be output by the processor 202 is stored in the memory 206 and the memory 206 also stores programs/codes for instructing the processor 202 to execute certain actions, which will be elaborated below, and data which the instructions operate on. The optimization engine 200 is configured to produce an optimized output including a modified project schedule (or schedules) having an optimal project makespan satisfying cost constraints or optimal project cost satisfying time constraints, for one or multiple projects as explained below.

Figure 3 is a flow chart illustrating a method 300 for deriving an optimal project makespan performed by the optimization engine 200. In this embodiment, a repair and maintenance project 250 (see Figure 7) of a vessel in a shipyard is used as an example although it would be appreciated that the method is also applicable for deriving an optimal project makespan for multiple projects.

At step 302, the optimization engine 200 receives definition of the repair and maintenance project 250 by a user. The project definition 302 also includes defining tasks to complete the repair and maintenance project 250 and the five tasks defined are found in Figure 7 which are:

T1 – docking of vessel;

T2 – maintenance of hull;

T3 – maintenance of ballast;

T4 – repair of steel plate;

T5 – Undocking of Vessel.

Actions 702,704,706,708,710 required for each tasks T1,T2,T3,T4,T5 are defined in Figure 7.

At step 304, a critical chain for the repair and maintenance project 250 is defined based on the defined tasks T1,T2,T3,T4,T5 and this involves assigning resources by the user (as identified from the resource constraint database 118) to complete each task, and leveling the resources so as to resolve resource conflicts between projects which require the same resource to complete a task based on Critical Chain Project Management (CCPM) methodology. The longest sequence of resource-leveled tasks for each project is identified as the critical chain 260 for that project.

Figure 7 also illustrates the resources allocated by the user to complete each task T1,T2,T3,T4,T5 for the project 250. In the case of the task T1, the resources allocated are “Dry Dock and dry dock personnel” 712; for the task T2, the resources allocated are “High Pressure Jet; 4 x painters etc” 714; for the task T3, two maintenance personnel 716 is allocated; for the task T4, the resources are “cutting machine, welding tools etc”; and for the undocking task, “Dry Dock and dry dock personnel” are allocated.. During the definition of the

critical chain at step 304, the user also identifies priorities for each task, if applicable and indicates against each resource whether additional resource such as overt-time or outsourcing of the task to subcontractors is available to compliment or replace the allocated resources. By default, all the tasks are assigned normal priority when each task is created and the user (e.g. project managers) may change the priority during the course of the project. Likewise, when the resource is first allocated to one of the task, the user then determines whether OT or outsourcing is allowable to be assigned to that task and performs the assignment. The assignment may take place during the resource allocation at step 304 or at any time prior to optimizing of the project.

Based on CCPM, the allocated resources are leveled and the critical chain 260 is identified and this is shown pictorially in Figure 5a with the tasks T1,T2,T3,T5 forming the critical chain 260 for the repair and maintenance project 250. It should be appreciated that the task T4 is performed in parallel with the other two tasks T2,T3, and the duration to complete the tasks T1,T2,T3,T4,T5 as identified in the critical chain 260 is defined as an initial project makespan 262.

The identification of the critical chain at step 304 may involve a preprocessing step, especially if the method 300 is configured as a plug-in computer program which interacts with a host project management software application such as Microsoft Project™ so that the method 300 is compatible with parameters of the host project management software application. The preprocessing may involve backing-up relevant project, task, and resource assignment data such as task duration etc; validating and adjusting project schedule constraints to meet

critical chain algorithm needs; and creating special project buffer task and project end milestone task for the whole project if necessary; exploiting the project and all its subprojects, external tasks/projects with the following changes:

(i) add project milestone task to individual subprojects and adjust task dependence(s) if necessary; and

(ii) change their tasks' planning policy to be "As Late As Possible" to meet critical chain algorithm needs.

At step 306, the optimization engine 200 checks whether there is budget control for the repair and maintenance project 250 based on user-defined inputs. If there is budget control, the optimization engine 200 validates whether budget defined by the user is correct or not at step 308 for each project and then goes to step 310. If there is no budget control, the optimization engine 200 proceeds directly to step 310.

At step 310, the optimization engine 200 identifies any critical task in the critical chain 260 which have yet to be assigned additional resources. A critical task is defined as a task which determines duration or makespan of a project. It must finish on time for the entire project to finish on time. If a critical task is delayed, the project completion date is also delayed. Any task in a Critical Chain is defined as a critical task. In the case of the critical chain 260 depicted in Figure 5a, the critical tasks which determines the initial project makespan 262 are tasks T1,T2, T3 and T5 and these set of tasks is identified as the critical tasks. Since task T4 does not affect the initial project makespan 262 of the critical

chain 260, the task T4 is not identified as a critical task in Figure 5a. At step 312, the optimization engine 200 checks to see if there is any critical task identified in step 310 and since four tasks T1,T2,T3,T5 have been identified as critical tasks, the method 300 proceeds to step 314.

At step 314, the optimization engine 200 assigns additional resource of Over-Time (OT) or subcontractor to each critical task T1,T2,T3,T5 if applicable, as defined earlier by the user. In this example, the tasks T1,T3 and T5 were not allocated or associated with any additional resource, and OT was assigned by the user as a possible additional resource for the task T2. Thus, the optimization engine 200 allocates or assigns OT to the task T2 as shown in Figure 5b.

At step 316, the optimization engine 200 re-defines the critical chain 260 in view of the assignment of OT to the task T2 and the effect of this is shown in Figure 5c which depicts shortening of the duration to complete the task T2 when compared with Figures 5a or 5b.

After step 316, the method 300 goes back to step 310 to identify any critical tasks in view of the redefinition of the critical chain 260 and this is depicted as Figure 5d which is a duplicate of Figure 5c. It should be apparent that the critical chain's makespan is now defined by the duration of the task T1,T4 and task T5 and thus, at step 310, these three tasks T1, T4 and T5 are identified as the critical tasks in this cycle.

Next the method 300 goes to step 312 and the optimization engine 200 notes that the task T1 and T5 have already been identified earlier (which may be by way of a remark against the task T1 and T5, or flag) and that the task T4 has yet to be checked, so the critical task list is not empty. Thus, the optimization engine 200 proceeds to step 314 and checks whether an additional resource is to be allocated to the task T4. In this respect, the user has assigned possibility of outsourcing the task T4 to a subcontractor and the optimization engine 200 allocates a subcontractor to take over or replace the originally allocated resource 718 in Figure 7 and this allocation is shown in Figure 5e.

The method 300 next goes to step 316 which re-defines the critical chain 260 and the effect is shown in Figure 5f which shows a reduction in duration of the time to complete the task T4 (when compared to Figures 5d and 5e). It should be apparent that the makespan of the repair and maintenance project 250 is dependent on the duration of the tasks T1,T4 and T5. With the critical chain 260 redefined at step 316, the method 300 goes back to step 310 again to identify any critical tasks which have yet to be allocated additional resources, and the optimization engine 200 notes that all the tasks T1,T2,T3,T4,T5 have been completed. Thus, at step 312, the optimization engine 200 notes that there is no critical task (i.e. it is empty) and the method 300 goes to step 318.

During step 314, in the allocation of additional resources such as OT or outsourcing, the optimization engine 200 also tracks two parameters:

a) "DurationChanges" which defines changes in duration for completing a task prior to and after allocating the predefined additional resource; and

b) "TotalCostChanges" which defines changes in project cost for completing a task prior to and after allocating the predefined additional resource.

It should be appreciated that the steps 310 to 316 are recursive and is performed until there is no new task (task that is not in the critical chain before) inserted into critical chain list by means of heuristics such as applying Over-Time or Subcontractor to a task if the duration of the task reduces based on the priorities of projects and tasks and heuristic rules as follows:

1) No change in cost or duration if no Over-Time and no Subcontractor;

2) If a task is to be sub-contracted out, then Over-Time does not apply;

3) Applying Over-Time to a task will shorten its duration. The cost may change;

4) Sub-contracting a task may change its duration. The cost may change.

Next, at step 318, the optimization engine 200 determines minimum project makespan 264 and minimum project cost based on heuristic rules that adjusts task durations and costs as described earlier and the following steps are performed:

1) The makespan of the repair and maintenance project 250 at step 316 which leads to no further critical tasks being identified at step 310 is considered to be the minimum project makespan 264 since the critical chain 260 of Figure 5f is the shortest after allocating the predefined additional resources to the applicable tasks T2,T4.

It should be appreciated that a project cost associated with the minimum project makespan 264 is not the minimum project cost in view of the allocation of additional resource such as OT or outsourcing which may lead to increase costs.

Next, the optimization engine 200 calculates the minimum project cost:

2) For each task associated with OT or outsourcing, assign the Over-Time or Subcontractor if applicable based on the heuristic rules described above for assigning Over-Time and Subcontractor to each task. As it can be appreciated from Figure 5, the critical chain is dynamic and changes whenever task duration changes due to allocation of additional resource such as OT or outsourcing at step 314 and there may be instances in which one of the tasks for the project 250 has been allocated with OT or outsourcing but is not part of the critical chain (which are called non-critical tasks). This means that cost changes due to the OT/outsourcing allocation have yet to be considered and thus, at step 318, the optimization engine 200 is configured to track the cost changes associated with allocation of such additional resources.

3) For each task with Over-Time assignment, compute the change of costs for each assignment and add it to TotalCostChanges of the task;

4) For each task with Subcontractor assignment, compute the change of costs for each assignment and add it to TotalCostChanges of the task;

5) Calculate TotalCostIncreased by sum up all TotalCostChanges > 0;

6) Calculate difference of the cost of the current project(s) and TotalCostIncreased as the minimal cost.

As explained above, the tasks explained above, may include the critical tasks (current and previous ones) or the non-critical tasks which are required to complete the project but do not form part of the critical chain since such non-critical tasks do not affect the project makespan.

Applying the above algorithm to Figure 5, let's assume that with OT allocation to the task T2, the increase in cost is $500. In the case of the task T4 in which outsourcing is assigned, this resulted in a reduction in the cost for completing the task T4, and the "TotalCostChanges" for assigning outsourcing to T4 is -$200 (negative sign to reflect cost saving).

Next in paragraph (4) above, the "TotalCostIncreased" is calculated by summing up "TotalCostChanges > 0". The cost changes which are more than zero is the allocation of OT which resulted in an increase of $500 to the project cost. Thus, the "TotalCostIncreased" for the project in Figure 5 is thus $500. Next, the cost difference is calculated to obtain the minimum project cost = current project costs – "TotalCostIncreased". Assuming that the current project cost (as explained earlier, this corresponds to the minimum project makespan of Figure 5f) is $X, the minimum project cost would thus be ($X-$500). Of course, the above is a simple example, but in reality, there may be hundreds or thousands of tasks (critical and non-critical tasks) to track the change in cost due to the allocation of additional resource.

Based on the minimum project makespan and minimum project cost calculated in step 318, step 320 checks whether the minimum project cost is more than the

budget. If the minimum project cost is more than the budget, the optimization engine 200 performs step 322 which is post processing and then the method 300 ends to highlight that the budget may not be feasible to complete the repair and maintenance project 250. If the minimum project cost is less than or equal to the budget, then the optimization engine 200 goes to step 324.

At step 324, the optimization engine 200 determines the current project cost which is the cost to complete the repair and maintenance project 250 based on the minimum project makespan illustrated in Figure 5f. At step 326, the optimization engine 200 compares the current project cost with the budget and if the current project cost is less than or equal to the budget, the optimization engine 200 executes step 332 to find the optimal project makespan, this being the minimum project makespan 264 identified in Figure 5f.

If the current project cost is more than the budget, the optimization engine 200 executes step 328 which un-assigns OT or outsourcing to subcontractors to reduce the current project cost and Figure 4 illustrates the steps to perform. At step 402, the optimization engine 200 sorts the assigned OT and subcontractor tasks by "TotalCostChanges" and "DurationChanges" in descending order. In the example of Figure 5f, the sorted task T2 would have a higher "TotalCostChanges" of S$500 than that for outsourcing task T4 which yields savings of –S$200. At step 404, the optimization engine 200 next un-assigns the OT or outsourcing based on the highest TotalCostChanges, and in this case, the OT is un-assigned from the task T2 since this cost more than the outsourcing of the task T4. With the un-assignment, the project's critical chain is

revised at step 406 and the duration of the task T2 would be similar to that shown in Figure 5b.

The method 300 next returns to step 330 of Figure 3 to calculate a new project cost and makespan due to the un-assignment of OT to the task T2. The new project cost is compared with the budget at step 326 and if it is not over the budget, the optimization engine 200 goes to step 332 which defines the new project makespan as the optimal makespan. If the new project cost is more than the budget, steps 328 to 330 are repeated.

It should be appreciated that the new project cost is due to the un-assignment of the OT which in other words, is exactly the same as the minimum project cost calculated at step 318 and thus, this new project cost would definitely be less than the budget. In other words, step 320 provides a gate to ensure that if the method 300 branches to step 324, in the worst case scenario, the un-assignment of the additional resources of OT or outsourcing would result in reducing the project cost to the minimum project cost.

After obtaining the optimal makespan at step 332, the method 300 goes to step 322 of Figure 3 to perform the post processing. The post processing includes:

(i) Setting Critical Chain flag (=true) to the critical tasks in the identified Critical Chain; or uncheck Critical Chain flag (=false) if the ex-critical task becomes non-critical;

(ii) Recording the task's duration change to its Notes as audit trial if it's changed in the Identify Critical Chain process. Notes are an explanatory text attached or associated with a task of the project.

(iii) Recording the project and its subprojects' duration change and initial cost to their Notes as audit trial;

(iv) Clearing backup information to release memory space;

(v) Hiding all the internal processing data and temporary files; and/or

(vi) Presenting the final result in a combinable custom view (e.g., customized Enterprise Gannt Chart with customized Resource View); for example as that shown in Figure 8. In this example, the optimal makespan for the repair and maintenance project 250 is derived to be 43 days which is a reduction from the initial project makespan of 60 days.

After post processing, the method of Figure 3 ends.

A sample GUI for implementation of the method described in Figure 3 is shown in Figure 6.

Figure 9 is a flow chart for a method 900 for obtaining an optimal project cost also performed by the optimization engine 200 using overtime or outsourcing strategies as additional resource. The method 900 will be described based on the example of the repair and maintenance project 250.

It should be appreciated that steps 902, 904, 910, 912, 914, 916, 918, 922, 928 and 930 of Figure 9 are similar to corresponding steps 302, 304, 310, 312, 314,

316, 318, 322, 328 and 330 of Figure 3 and thus, the above explanations are applicable for Figure 9.

At step 906, instead of budget control in step 306, step 906 checks whether there is duration or time limit control. If there is duration limit control, the method of Figure 9 proceeds to validate a predefined duration limit at step 908 (instead of validating budget in step 308 of Figure 3) and the optimization engine 200 checks that the duration limit input by the user is correct. After validation or if there is no duration limit control at step 906, the method branches to step 910 to identify critical task, if any, in the critical chain similar to that in step 310 of Figure 3.

Steps 910 to 916 are performed similar to steps 310 to 316 of Figure 3 and the end result is that shown in Figure 5f, which depicts the minimum project makespan 264 and the minimum project cost is also obtained just like what is described earlier at step 318 of Figure 3 for obtaining the optimal project makespan.

At step 920, the optimization engine 200 checks whether the minimum project makespan is more than the duration limit. If the minimum project makespan is more than the duration limit, the method goes to step 922 for post-processing (refer to the explanation of step 322 discussed earlier) since the budgeted duration limit may not be feasible to complete the project 250. If the minimum project makespan is not more than the duration limit, step 924 is performed to

determine the current project makespan based on the minimum project cost obtained at step 918.

At step 926, the optimization engine 200 compares the current project makespan with the duration limit and if the current project makespan is less than or equal to the duration limit, the method 900 goes to step 932 which defines the minimum project cost as the optimal project cost for the project 250.

If the current project makespan is over the duration limit, at step 928, the optimization engine 200 un-assigns OT or outsourcing to subcontractors similar to step 328 of Figure 3. At step 928, the optimization engine 200 performs the steps in the flow chart of Figure 10 which is similar to the flow chart of Figure 4. However, instead of sorting "TotalCostChanges" and "Duration changes", at step 1002 the optimization engine 200 sorts by "DurationChanges" to determine which assigned OT/outsourcing results in a highest change in duration. Of course, it is envisaged that the sort may be based on both parameters, just like Figure 4. Taking the example of Figure 5 and if the change in duration of the task T4 is larger after assigning the outsourcing than the change in duration of the task T2 after assigning OT, then at step 1004, the optimization engine 200 un-assigns the outsourcing to the task T4. With this un-assignment, the project's critical chain is updated at step 1006 and pictorially, this is similar to the critical chain illustrated in Figures 5d or 5e. Based on this revised critical chain, at step 930, the new project cost and makespan is calculated and the new project makespan is compared with the duration limit at step 926. If the new project makespan is less than or equal to the duration limit, the method 900

goes to step 932 and defines the new project cost associated with the new project makespan as the optimal project cost for the project 250.

Just like Figure 3, if at step 926, the new project makspan is still over the duration limit, steps 928 and 930 are repeated.

The method of Figure 9 ends after the post processing step 922, which performs functions similar to step 322 of Figure 3. A sample GUI for implementation the method illustrated in Figure 9 is shown in Figure 11.

As it can be appreciated, the proposed methods are capable of handling one or multi-project scheduling and able to derive optimal makespan and/or costs by assigning additional resource such as overtime or outsourcing of tasks.

In terms of management of multiple projects, the methods of Figure 3 and 9 may be configured to optimize project makespan or cost for multiple projects. For example, Figure 12a shows a project schedule for three projects – Project 1, Project 2 and Project 3 prior to optimizing their makespans, and Figure 12b shows the project schedule after optimization. Through applying the method proposed in Figure 3, it can be appreciated that the three project's makespan have been optimized so that an overall optimal makespan is also obtained for all the three projects.

The described embodiment is not to be construed as limitation. For example, although the method for optimizing makespan in Figure 3 is described separate

from the method for optimizing costs in Figure 9, it should be apparent that the features in one method may also be applied in the other method. Further, the pre-processing and post-processing steps described in Figures 3 and 9 may not be necessary and may be omitted depending on application. Further, leveling of resources may not be carried out if there is no conflict in the resources to be allocated to the tasks of a project.

At step 402 of Figure 4, the sorting was performed based on parameters "TotalCostChanges" and "DurationChanges" but it is envisaged that the sorting need only be performed based on one parameter and in the example of Figure 4, sorting based on "TotalCostChanges" is sufficient.

In the described embodiment, the additional resource allocated for the task T4 in Figure 5e is to outsource the task to a subcontractor which replaces the originally assigned resource 718 of Figure 7. However, it is envisaged that outsourcing of a task may compliment the originally assigned resource such as Dry Dock, rather than replace it.

The described methods may be implemented as a standalone software program or as a plug-in computer program to interact with host project management software such as MS Project™.

In the described embodiment, the user assigns either OT or outsourcing to a subcontractor as an additional or alternative resource but it is envisaged that the user may assign both or more additional resource to a task. In this way,

optimization engine 200 may select which additional resource to select based on greater cost or change in duration. Also, certain additional resources may be subjected to certain restrictions for example, it may be stipulated by law that a worker may not work more than a certain numbers of OT, and the method may take this into consideration when assigning the amount of OT so that it is within the legal limit.

The described embodiment uses the repair and maintenance project 250 in a shipyard as an example but it is envisaged that the described embodiment is also applicable to other industries which manages projects such as the construction industry, healthcare (such as in hospitals in which operating rooms, equipment, specialist etc are shared), manufacturing such as car manufacturing, or in the aerospace industry such as maintenance, repair and operations of aircrafts or flights, and even projects within an organization.

Having now fully described the invention, it should be apparent to one of ordinary skill in the art that many modifications can be made hereto without departing from the scope as claimed.

CLAIMS

1. A computer implemented method of obtaining an optimal project makespan, comprising

(i) receiving a plurality of tasks for completing the at least one project within an initial project makespan, identifying at least one critical task from the plurality of tasks which determines the initial project makespan, deriving a critical chain for the at least one project based on the at least one critical task;

(ii) allocating additional resource to the at least one critical task to obtain a minimum project makespan of the at least one project;

(iii) deriving a current project cost based on the minimum makespan;

(iv) if the current project cost is equal or less than a predetermined budget, identifying the minimum makespan as the optimal project makespan;

(v) if the current project cost is more than the predetermined budget, adjusting the allocation of the additional resource to the at least one critical task to derive a project cost which is equal or less than the budget; the project cost being associated with a revised project makespan; and

(vi) identifying the revised project makespan as the optimal project makespan for the at least one project.

2. A computer implemented method of obtaining an optimal project cost, comprising

(i) receiving a plurality of tasks for completing the at least one project within an initial project makespan, identifying at least one critical task from the plurality of tasks which determines the initial project makespan, deriving a critical chain for the at least one project based on the at least one critical task;

(ii) allocating additional resource to the at least one critical task to obtain a minimum project makespan of the at least one project;

(iii) calculating a minimum project cost based on the allocation of the additional resource;

(iv) deriving a current project makespan for the project based on the minimum project cost;

(v) if the current project makespan is equal to or less than a predetermined duration limit, identifying the minimum project cost as the optimal project cost;

(vi) if the current project makespan is more than the predetermined duration limit, adjusting the allocation of the additional resource to the at least one critical task to derive a project makespan which is equal or less than the predetermined duration limit; the project makespan being associated with a revised project cost; and

(vii) identifying the revised project cost as the optimal project cost for the at least one project.

3. A method according to claim 1, wherein prior to step (iii), the method comprises calculating a minimum project cost for the at least one project; and comparing the minimum project cost with the budget.

4. A method according to claim 3, wherein if the minimum project cost is more than the budget, the method further comprises outputting that the budget is not able to complete the at least one project.

5. A method according to claim 4, wherein there is more than one critical task assigned with the additional resource, and step (ii) includes tracking change in cost after allocation of the additional resource to the critical tasks.

6. A method according to claim 5, further comprising summing up all the change in cost to derive a total change in cost; and deducting the total change in cost from the current project cost to obtain the minimum project cost.

7. A method according to claim 5 or 6, wherein step (v) includes un-allocating the allocation of the additional resource to derive the project cost which is equal or less than the budget.

8. A method according to claim 7, wherein the un-allocation includes arranging the critical tasks in order of the tracked change in cost; each critical task being associated with a said additional resource; and un-allocating the said additional resource one at a time to derive the project cost which is equal or less than the budget.

9. A method according to any of claims 5 to 8 when dependent on claim 6, wherein the plurality of tasks includes non-critical tasks, at least some of which are associated with additional resources; and the method comprises tracking change in cost after allocation of the additional resources to the non-critical tasks, and adding the tracked change in cost of the non-critical tasks to the total change in cost.

10. A method according any of claims 1, and 3 to 9, wherein step (i) includes deriving a critical chain for multiple projects, and the method includes obtaining the optimal project makespan for each of the multiple projects or for all the multiple projects.

11. A method according to claim 2, wherein step (ii) comprises comparing the minimum project makespan with the duration limit.

12. A method according to claim 11, wherein if the minimum project cost is more than the duration limit, the method further comprises outputting that the duration limit is not able to complete the at least one project.

13. A method according to claim 12, wherein there is more than one critical task assigned with the additional resource, and step (iii) includes tracking change in cost after allocation of the additional resource to the critical tasks.

14. A method according to claim 13, further comprising summing up all the change in cost to derive a total change in cost; and deducting the total change in cost from the current project cost to obtain the minimum project cost.

15. A method according to claim 13 or 14, wherein step (vi) includes un-allocating the allocation of the additional resource to derive the project makespan which is equal or less than the duration limit.

16. A method according to claim 15, wherein the un-allocation includes arranging the critical tasks in order of the tracked change in duration; each critical task being associated with a said additional resource; and un-allocating the said additional resource one at a time to derive the project makespan which is equal or less than the duration limit.

17. A method according to claim 14, wherein the plurality of tasks includes non-critical tasks, at least some of which are associated with additional resources; and the method comprises tracking change in cost after allocation of the additional resources to the non-critical tasks, and adding the tracked change in cost of the non-critical tasks to the total change in cost.

18. A method according any of claims 2, and 11 to 17, wherein step (i) includes deriving a critical chain for multiple projects, and the method

includes obtaining the optimal project cost for each of the multiple projects or for all the multiple projects.

19. A method according to any preceding claim, wherein the additional resource include overtime and/or outsourcing of tasks to subcontractors.

20. A method according to any preceding claim, wherein if the additional resource allocated to the at least one critical task includes outsourcing of tasks to subcontractors, the outsourcing of the tasks replaces an initial resource allocated to the at least one critical task under step (i).

21. A method according to any preceding claim, wherein step (i) includes assigning constrained resources shared with other projects to the plurality of task, and leveling the constrained resources.

22. A method according to any preceding claim, wherein prior to step (i), receiving user definition of the plurality of task required for completing the at least one project and initial resources for allocation to each of the task for deriving the critical chain.

23. A method according to any preceding claim, wherein step (i) includes a preprocessing step to adapt the method to be compatible with a third party project management application.

24. A method according to any preceding claim, further comprising receiving task priorities, and associating the task priorities with each one of the plurality of task.

25. A method according to any preceding claim, wherein step (i) includes (a) identifying a set of critical tasks for the or new critical chain, at least some of which are associated with additional resources; and step (ii) includes

(b) checking if any of the critical task's duration has not been adjusted due to the associated additional resource, and if there is, identifying such critical tasks;

(c) adjusting the duration of such critical tasks based on the associated additional resource;

(d) deriving a new critical chain based on the adjusted duration; and

(e) repeating steps (a) to (d) until the duration of all the critical tasks associated with the additional resource have been adjusted to derive a revised critical chain, and obtaining the minimum makespan based on the revised critical chain.

26. A plug-in computer program product comprising instructions when executed are arranged to control a computer to perform the method according to any preceding claim.

27. A computer program product stored on a data recording medium comprising instructions when executed are arranged to control a computer to perform the method according to any of claims 1 to 25.

28. Apparatus for obtaining an optimal project makespan, comprising a processor configured to

(i) receive a plurality of tasks for completing the at least one project within an initial project makespan, identify at least one critical task from the plurality of tasks which determines the initial project makespan, derive a critical chain for the at least one project based on the at least one critical task;

(ii) allocate additional resource to the at least one critical task to obtain a minimum project makespan of the at least one project;

(iii) derive a current project cost based on the minimum makespan;

(iv) if the current project cost is equal or less than a predetermined budget, identify the minimum makespan as the optimal project makespan;

(v) if the current project cost is more than the predetermined budget, adjust the allocation of the additional resource to the at least one critical task to derive a project cost which is equal or less than the budget; the project cost being associated with a revised project makespan; and

(vi) identify the revised project makespan as the optimal project makespan for the at least one project.

29. Apparatus for obtaining an optimal project cost, comprising a processor configured to

(i) receive a plurality of tasks for completing the at least one project within an initial project makespan, identify at least one critical task from

the plurality of tasks which determines the initial project makespan, derive a critical chain for the at least one project based on the at least one critical task;

(ii) allocate additional resource to the at least one critical task to obtain a minimum project makespan of the at least one project;

(iii) calculate a minimum project cost based on the allocation of the additional resource;

(iv) derive a current project makespan for the project based on the minimum project cost;

(v) if the current project makespan is equal to or less than a predetermined duration limit, identify the minimum project cost as the optimal project cost;

(vi) if the current project makespan is more than the predetermined duration limit, adjust the allocation of the additional resource to the at least one critical task to derive a project makespan which is equal or less than the predetermined duration limit; the project makespan being associated with a revised project cost; and

(vii) identify the revised project cost as the optimal project cost for the at least one project.