



Effective Project Scheduling Under Workspace Congestion and Workflow Disturbance Factors

Vitaly Semenov, Anton Anichkin, Sergey Morozov, Oleg Tarlapan, Vladislav Zolotov (Institute for System Programming RAS, Moscow, Russia)

Abstract

Effective project management implies the use of advanced planning and scheduling methods that allow to determine feasible sequences of activities and to complete a project on time and on budget. Traditional scheduling tools like fundamental Critical Path Method (CPM) and various methods for Resource Constrained Project Scheduling Problem (RCPSP) and Time Constrained Project Scheduling Problem (TCPSP) have many shortcomings for construction projects where spatial factor plays a critically important role. Previous attempts to interpret space as a specific resource were successful for particular problems of line-of-balance scheduling, space scheduling, dynamic layout planning, horizontal and vertical logic scheduling, workspace congestion mitigating, scheduling multiple projects with movable resources, spatial scheduling of repeated and grouped activities and motion planning. However, none of these methods considers the spatio-temporal requirements in a holistic framework of generic RCPSP problem and provides feasible results accounting for workspace and workflow factors. In this paper we start with the classical RCPSP statement and then present mathematically strong formalisation of the extended generalised problem, taking into account workspace congestion and workflow disturbance constraints specified in practically meaningful and computationally constructive ways. For the generalised RCPSP problem an effective scheduling method is proposed. The method tends to minimise the project makespan while satisfying timing constraints and precedence relations, not exceeding resource utilisation limits, avoiding workspace congestions and keeping workflows continuous. The method reuses so-called serial scheduling scheme and provides for additional computational routines and heuristic priority rules to generate feasible schedules satisfying all the imposed requirements. Advantages of the method and prospects for its application to industrial needs are outlined in the paper too.

Keywords: Planning and scheduling, Resource-constrained project scheduling problem, Priority rules, 4D modelling, Workspace management.

Introduction

Effective project management implies the use of advanced planning and scheduling methods that allow to determine feasible sequences of activities and to complete a project on time and on budget. Critical Path Method (CPM) and various methods for Resource Constrained Project Scheduling Problem (RCPSP) and Time Constrained Project Scheduling Problem (TCPSP) are traditional tools incorporated in most popular project management systems like Microsoft Project, Oracle Primavera, Asta Powerproject.

Developed in the 1950s, the CPM generates useful information about the project, such as the longest sequence of activities, the shortest project duration, and the total and free floats of each activity. This information is crucial to a project's success and substantially important for the project manager to plan and control it more actively and efficiently. In the main, critical activities having zero floats should receive the management attention that might be unnecessary on other activities. This management by exception is an important advantage of the CPM, especially on large, complex projects (Ahuja, 1976; Bowers, 1995). Later the original CPM formulation was generalised to take into account resource limitations within the RCPSP and TCPSP statements. In most of real industrial projects, scheduling without considering these limitations may lead to non-credible results, since the execution of activities is strongly affected by resource availability. Various analytical and heuristic methods have been developed to apply the resource availability into the scheduling process

(Ahuja, 1976; David and Patterson, 1975; Hegazy, 1999). Analytical methods attempt to find the optimum solution in terms of the minimum project duration, but usually require very long computational time, making them impractical. On the other hand, heuristic approaches provide reasonable solutions for large-scale projects in practical time (Boctor, 1990; Hegazy, 1999).

However, these methods ignore divergent spatial factors and cannot guarantee the correctness of the prepared schedules in terms of lack of spatial conflicts commonly related to workspace congestion and workflow disturbance. Indeed, an activity can be performed if only all the needed workspaces are reserved throughout its execution period and if they are not occupied by other competitive activities arranged at the same place at the same time. In some sense, workspaces can be interpreted as renewable resources shared among concurrent project activities with predefined utilisation rates. This observation applies equally to spaces required to install or to assemble product components, to store materials on logistics sites, spaces used as passageways to deliver resources to destination areas or reserved for parking zones or household rooms, and spaces preventing safety hazards. The workflow disturbance is another factor preventing prompt movement of resources on a project site, increasing idle time for labour and equipment, and thereby, deteriorating their productivity. To allow for cost and time efficiencies, it is necessary to achieve workflow continuity by balancing the resource utilisation and replacement.

Many researchers addressed these topics by means of the introduced concepts of line-of-balance (LOB) scheduling (Pai et al, 2013), space scheduling (Choo and Tommelein, 1999), dynamic layout planning (Zouein and Tommelein, 1999), horizontal and vertical logic scheduling (Thabet and Beliveau, 1994b), workspace congestion mitigating (Yeoh and David, 2012), scheduling multiple projects with movable resources (Hegazy, 1999), spatial scheduling of repeated and grouped activities (Thabet and Beliveau, 1994a), and motion planning (Ellips and Davoud, 2007). However, these attempts were successful only for very particular statements as well as did not result in a holistic framework accounting for workspace and workflow factors and extending traditional CPM, RCPSP and TCPSP methods.

In particular, LOB is a linear scheduling method that allows balancing of the operations in the projects with repeated activities continuously performed in each consecutive unit. Repeating units are commonly found as typical floors in high rise buildings, residences in multi-housing developments, stations in highways, meters in pipeline network, long bridges, tunnels, railways, or water mains. Using LOB, repetitive activities are scheduled in such a way to ensure a smooth procession of resources from unit to unit with minimal conflicts. However, many researches indicated that this technique is suitable to model simple repetitive production processes, but it is quite limited for the complex projects represented by discrete activities with varied utilisation and productivity rates.

In this paper, alternative scheduling formulations are discussed to extend the classical CPM and RCPSP statements and to account for workspace and workflow factors. In Section 2 we start with the classical statements and then provide mathematically strong formalisation of the generalised problem with the workspace congestion and workflow disturbance constraints specified in practically meaningful and computationally constructive ways. An effective scheduling method for the generalised problem is presented in Section 3. The method tends to minimise the project makespan while satisfying timing constraints and precedence relations, not exceeding resource utilisation limits, avoiding workspace congestion and keeping workflows continuous. The method reuses so-called serial scheduling scheme and provides for additional computational routines and heuristic rules to generate feasible schedules satisfying all the imposed requirements. Section 4 is devoted to

preliminary validation of the method. In the conclusions section, advantages of the method are summarised and prospects for its application in industrial practice are outlined.

Generalised Scheduling Problem

Classical RCPSP formulation

The classical RCPSP problem can be stated as follows. A single project can be represented by a network with N activities on the nodes and M links on its arcs. Every activity a_n , $n=1, \dots, N$ implies an uninterrupted process beginning at the time t_n and having the fixed duration $d_n \geq 0$. Every link l_m , $m=1, \dots, M$ reproduces the finish-start precedence relation between a predecessor activity $a_{Pr(m)}$ and a successor activity $a_{Sc(m)}$ and forces the successor activity not to be started earlier than the given lag τ_m after its predecessor has been finished. A successor activity having only zero-lag links cannot start until all its predecessors have been finished. For the formalisation unique dummy source and sink activities a_1 and a_N of zero duration $d_1=0$, $d_N=0$ are introduced and they are linked with the project activities having opened starts and opened ends respectively. In order to be processed, an activity a_n may require u_{nk} units of the renewable resource r_k during its execution. A constant availability of every resource r_k , $k=1, \dots, K$ is assumed and denoted as U_k . In correctly scheduled plan it cannot be exceeded at any time point t such that $t_1 \leq t \leq t_n$ throughout the whole project. In order to make the problem simple, activity splitting and resource levelling are not considered. The objective of the RCPSP is to schedule the activities such that the makespan of the project is minimised, all the precedence relations are satisfied and resource availability limits are not exceeded. Let $A(t)$ denotes an index set of the activities being in progress at the time t or formally $A(t) = \{n \mid n=1, \dots, N, t_n \leq t < t_n + d_n\}$, then the RCPSP problem can be mathematically formulated for unknown variable $X = \{t_n\}_{n=1}^N$ as follows:

$$\min t_N \text{ subject to} \quad (1)$$

$$t_{Sc(m)} \geq t_{Pr(m)} + d_{Pr(m)} + \tau_m, \forall m=1, \dots, M \quad (2)$$

$$\sum_{n \in A(t)} u_{nk} \leq U_k, \forall k=1, \dots, K, \forall t \mid t_1 \leq t \leq t_n \quad (3)$$

The objective function (1) minimises the completion time of the unique sink activity t_N and thereby the makespan of the whole project. Constraints (2) take into consideration the links between each pair of preceding and succeeding activities. Finally, constraints (3) limit the total resource utilisation at each time point to the available amounts. To be correct from mathematical point of view and to guarantee the solution existence, the RCPSP must avoid any link cycles and exclude exceeded resource utilisation for individual activities so that $u_{nk} \leq U_k, \forall n=1, \dots, N \forall k=1, \dots, K$.

By relaxing the resource constraints (3), the RCPSP reduces to the CPM-case which can be solved by forward recursion in polynomial time. But in general statement the RCPSP belongs to the class of NP-hard problems (Kolisch et al, 1995; Lavallo, 2006). Existing dynamic programming procedures as well as the branch and bound techniques are too computationally expensive to find optimal solutions in most practical cases. Therefore, heuristic approaches, and in particular, priority rule based scheduling methods, are usually employed within commercial packages for such purposes. Generally, such methods distinguish in a scheduling scheme (serial or parallel, single- or multi-pass) and in a set of

rules to prioritise the concurrent activities which over-consume the limited resources. Well-known priority rules are most total successors (MTS), latest start time (LST), greatest rank positional weight (GRPW), weighted resource utilisation ratio and precedence (WRUP), latest finish time (LFF), minimum slack (MSLK). Being combined and implemented within multi-pass schemes, they show the best results obtainable by heuristics today. For more details, please see (Kolisch et al, 1995).

Workspace Congestion Conditions

As stated above, the RCPSP only takes into account constraints for renewable resources. It can be extended by introducing workspaces that allow explicit visual interpretation and mathematically strong formalisation. Let $w_i, i=1,\dots,I$ are project workspaces geometrically represented by solids being connected, compact, orientable 3-dimensional manifolds in Euclidean space. Typically they are the objects of simple shape: cuboids, cylinders, prisms, pyramids, spheres, cones, polyhedra. But they can be compound objects constructed from primitives by means of Boolean operations on sets: union, intersection and difference. Being adopted by the constructive solid geometry modelling (CSG), these operations are traditionally denoted as \cup, \cap and \setminus respectively. Workspaces can overlap each other in different dimensions and across time and therefore they cannot be considered as independent resources.

By consuming u_{nk} units of the resource r_k with corresponding spatial rate v_k and operational time d_{nk} , the activity a_n utilises a workspace $w_{i(n,k)}$ with the factor

$$\rho_{nk}(t) = \begin{cases} u_{nk} \frac{v_k}{v(w_{i(n,k)})} \frac{d_{nk}}{d_n} & \text{if } t_n \leq t < t_n + d_n \\ 0 & \text{otherwise} \end{cases}, \quad (4)$$

where the function $v(w)$ returns the volume of the corresponding workspace. The introduced factor can be interpreted as an averaged density of the resource units per unit volume per unit time. A spatial multiplier in the expression gives a ratio of the space required by the resource unit to the total available space allocated to the activity. A temporal multiplier reflects the fact that workspaces may not always be utilised throughout the activity's operation time and may be used to describe the intermittent nature of continuous activities. A notation $w_{i(n,k)}$ is used here to emphasise that the workspace w_i is associated with the activity a_n and the related resource r_k only when the activity performs.

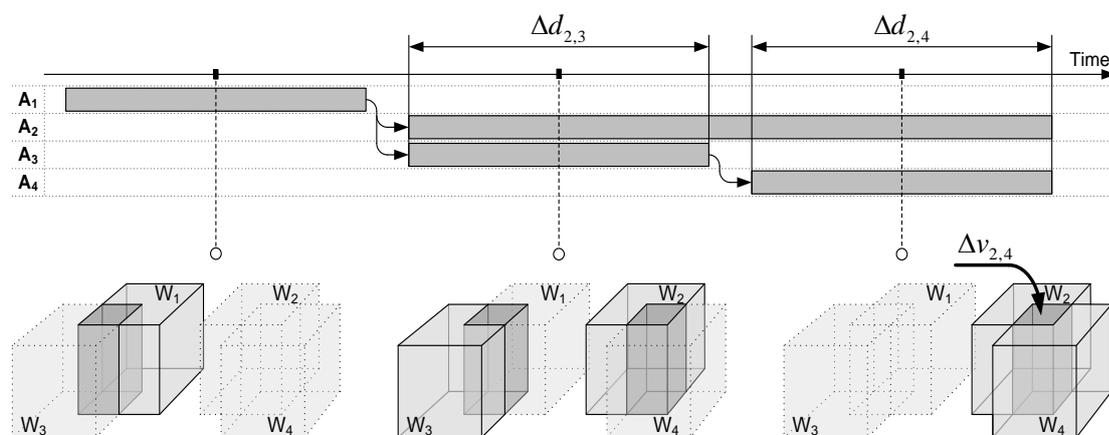


Figure 1: Example of workspace competition

Two workspaces $w_{i(n,k)}$, $w_{i'(n',k')}$ are defined as interfering ones if their originating activities overlap in a time interval $\Delta d_{n,n'} > 0$ and their solids intersect in a volume $\Delta v_{i,i'} = v(w_{i(n,k)} \cap w_{i'(n',k')}) > 0$. Then the workspace interference can be quantified by multiplying these factors. To be processed concurrently the activities must avoid workspace competition and congestion. By limiting the utilisation and congestion factors we thus require that the workspace capacity must be large enough to allocate all the needed amount of the resource units, including units allocated in other interfering workspaces. In other words, the conflicting activities should have an opportunity to be rearranged so that the utilised resources can be reallocated over free domains of the workspaces. Under the suggestion that workspaces are consumed by different activities and resources additively, this requirement takes the following form:

for $\forall i(n,k) \in I(A,U)$, $\forall t | t_n \leq t < t_n + d_n$

$$\sum_{(n',k') \in I(A,U)} \rho_{n'k'}(t) v(w_{i(n,k)} \cap w_{i'(n',k')}) \Delta d_{n,n'} \leq v(w_{i(n,k)}) d_n, (5)$$

where $I(A,U)$ is a set of index pairs for all the activities and related resources so that $u_{nk} \neq 0$. The constraint (5) is stated for every workspace $w_{i(n,k)}$ throughout the execution interval of its originating activity $t_n \leq t < t_n + d_n$. A summation on the left side of the constraint is taken over all the project workspaces, including the given workspace. It is essential that the constraints (5) allow short-term intersecting or even overlapping of workspaces on the condition that their utilisation and congestion factors are small enough.

Consider a sample schedule consisting of four activities A_1 – A_4 , each of them utilises own workspace W_1 – W_4 correspondingly. The workspaces are represented by solids having simple box shape and being located as shown by Figure 1. In spite of solids of the workspaces W_1 , W_3 as well as W_2 , W_4 intersect, the figure demonstrates the only case of competition and potential congestion of the workspaces W_2 , W_4 introduced by the activities A_2 and A_4 overlapping in the time interval $\Delta d_{2,4} = [t_4, t_4 + d_4]$. The workspaces W_1 , W_3 never interfere each other as the associated activities A_1 , A_3 are not performed concurrently according to the schedule. To detect if the activities A_2 , A_4 should be rearranged the additional analysis of utilisation and congestion factors for the workspaces W_2 , W_4 is required.

Often, the operational factor is removed from the consideration and, thereby the resources are suggested to be utilised throughout the whole activity duration. Then, the constraints take the following simplified form:

$$\text{for } \forall i(n,k) \in I(A,U) \quad \sum_{(n',k') \in I(n,k)} u_{n'k'} \frac{v_{k'}}{V_{(n',k')}} v(w_{i(n,k)} \cap w_{i'(n',k')}) \leq v(w_{i(n,k)}) \quad (6)$$

A summation on the left side of the constraint is taken over all the workspaces $I(n,k) \subseteq I(A,U)$ interfering with the given workspace $w_{i(n,k)}$. In order to guarantee the existence of a solution, the constraints (5) and (6) being applied to workspaces occupied by every individual activity must be satisfied.

Noteworthy, different models for quantifying spatio-temporal interference between workspaces have been proposed (Yeoh and David, 2012). The formalized constraint (5) is quite similar to the model discussed in (Chua et al, 2010), while the form (6) is more close to the criteria presented in (Chavada et al, 2012). Indeed, if three activities A_1 , A_2 , A_3 utilise the same resource r in the same workspace w throughout the same duration with respective

factors ρ_1, ρ_2, ρ_3 , the workspace mitigation requirement takes the trivial form $\rho_1 + \rho_2 + \rho_3 \leq 1$. The general forms (5), (6) are intended for more sophisticated cases when concurrent activities partially overlap in time and utilise different resources allocated in different, partially crossed workspaces.

Main disadvantage of the reduced representation (6) compared with the general form (5) is that even a short time overlay of the workspaces with relatively high utilisation factors may lead to a breach of the congestion conditions although unlikely that such workspace conflicts could not be resolved in practice. Nevertheless, in this paper the form (6) is used as more solid requirement imposed upon the interfering workspaces. Moreover, it requires less computation which is especially attractable for scheduling of large projects.

Workflow Disturbance Conditions

Workflow continuity is another important factor affecting the schedule feasibility and its practical value. Being arranged in different places (sometimes, in different sites or even in geographically remote regions) the project activities need the resources to be reallocated and replaced in proper workspaces. Regardless of how resources are moved, these factors add on project costs and duration inevitably. Ignoring these factors, the methods usually generate schedules with high resource traffic and unreasonable discontinuous workflows what makes them useless for practical purposes. This is a serious shortcoming of classical CPM, RCPSP, TCPSP methods for properly modelling the real-world constraints.

Unfortunately, the studies mentioned above did not result in common vision on the workflow phenomenon and did not provide a solid basis to specify workflow constraints in a formal way. In this section own model for workflow management is presented being tightly connected with issues of the spatio-temporal allocation of resources among workspaces. The model assumes the permanent use of a global pool of resources and local resource pools associated with separate workspaces. The global project pool stores the total amount of units for each resource type available at the current time. Local pools store similar amounts of units assigned to every workspace individually. At the initial time moment all the resource units are assigned to one or more workspaces emulating logistics sites, warehouses, parking zones or household rooms. Whenever a new activity starts and requests a fixed number of resource units, it should be taken into account not only the availability of the required units at the given time moment, but also their distribution over workspaces. If the requested units are available, the key issue arisen here is which workspaces the resources should be supplied from. Once the decision is made, global and local pools are updated properly so that the total amount of the resource units is decreased by the utilised amount. When the activity is complete, these units are released and placed in the same workspace where they have been utilised by the activity before. As they become available for other activities, the global pool and local pool of this workspace are updated so that the total amount of the available resource units is increased by the released amount.

It is seen the following tight relationship between resource flows and workflows in the scope of the model above. By supplying resource units from the nearest workspaces and minimizing resource reallocation time, the workflows become more regular. The reallocation processes can be simulated by additional links between the activities releasing and consuming the same resource units. Lag of every such link could be determined by means of an user-defined transfer function $\tau^k(\Delta u, i', i'')$ that returns the time needed to move Δu units of the resource r_k from the source workspace $w_{i'}$ to the destination workspace $w_{i''}$. If the activity a_n requires u_{nk} units of the renewable resource r_k and these units can be delivered from the workspaces of the finished activities $a_{n_1}, a_{n_2}, \dots, a_{n_M}$ so that

$u_{nk} = \Delta u_{nk}^{n_1} + \Delta u_{nk}^{n_2} + \dots + \Delta u_{nk}^{n_M}$, a structure $\bar{u}_{nk} = \left\{ \Delta u_{nk}^{n_m} \right\}_{m=1}^{M(n,k)}$ is called the route for the resource r_k in conformity to the activity a_n . Then additional transfer links should be created with the conditions below:

$$\text{for } \forall n = 1, \dots, N, \forall k = 1, \dots, K, \forall m = 1, \dots, M(n, k) \quad t_n \geq t_{n_m} + d_{n_m} + \tau^k(\Delta u_{nk}^{n_m}, i(n_m, k), i(n, k)) \quad (7)$$

Unfortunately, such links cannot be defined by the planner in advance in the problem statement phase and need to be determined directly when the project is being scheduled. Every created transfer link may delay the successor activity and therefore adds on overall project duration. The scheduling methods for thus formalized problem should take into account these circumstances when deciding on the activity priority minimizing the total project makespan and on the resource reallocation policy not disturbing natural workflows.

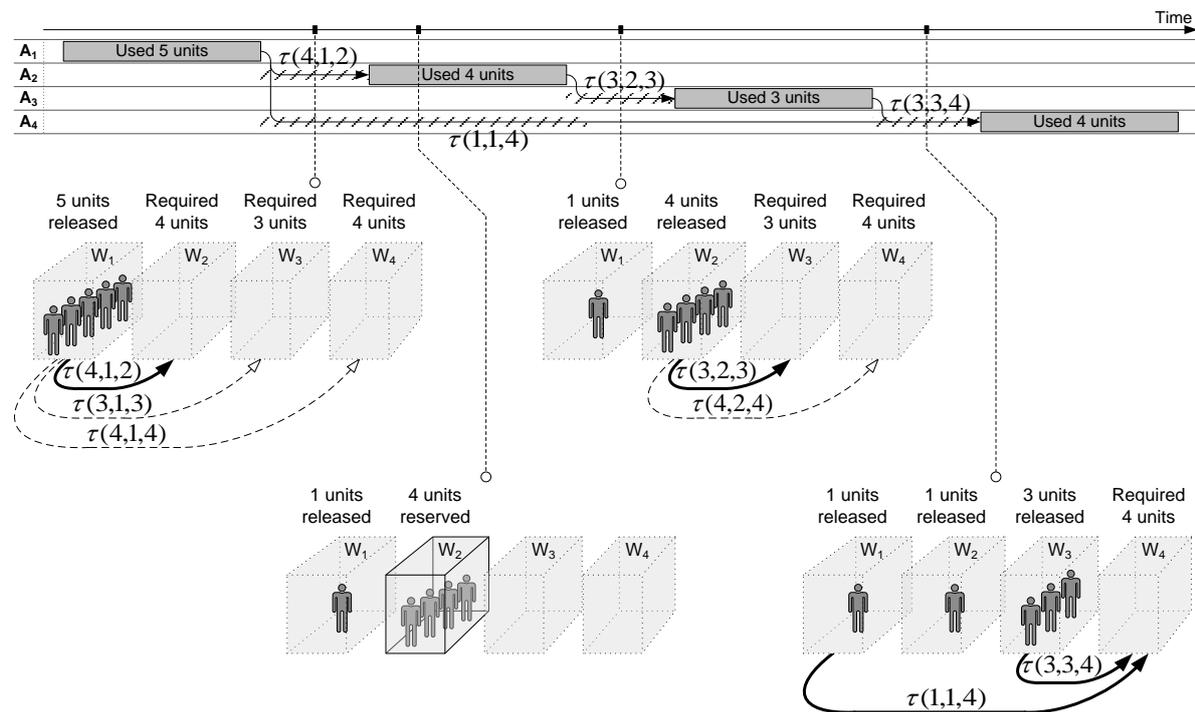


Figure 2: Example of a schedule taking into account workflow continuity factor.

Figure 2 presents an example of a schedule taking into account workflow continuity factor. The schedule consists of four activities A_1 – A_4 utilizing non-intersecting workspaces of simple box shape W_1 – W_4 correspondingly and requiring for execution the given amount of units of a labour resource "worker" as shown over the activity symbols at the Gantt chart. Let a crew consisting of five workers is available to perform activities on the schedule. The work starts with the activity A_1 using all the crew (5 units) that is located at the workspace W_1 . After the activity A_1 has been completed all the resource units become available for other activities of the schedule. Analysis of time values returned by transfer functions $\tau(4,1,2)$, $\tau(3,1,3)$, $\tau(4,1,4)$ determines the minimum as $\tau(4,1,2)$ moving four workers from W_1 to the closest workspace W_2 . Then the precedence relationship between the activities A_1 and A_2 is established so that A_2 should be started just after A_1 has been finished. A_2 reserves 4 resource units moved to the workspace W_2 . 1 resource unit has been

left released and located in the workspace W_1 . No more activities can be started concurrently with A_2 because of lack of sufficient free resource units to perform them.

Similarly, the activity A_3 is scheduled after A_2 has been completed. Three workers necessary to perform it are reserved and moved to the workspace W_3 . Two workers (one in W_1 , another one in W_2) are released, but this is not enough to start the activity A_4 concurrently. Finally, after the activity A_3 has been completed, A_4 requiring 4 resource units can be started. Three units are moved from the closest workspace W_3 , one more unit can be moved either from the workspace W_1 or W_2 . In spite of the workspace W_1 is outermost to W_4 , the resource unit located in it has been released earlier, so the time value $\tau(1,1,4)$ is less than $\tau(1,2,4)$. Then two precedence relationships between the activities A_1 and A_4 as well as A_3 and A_4 should be defined in the schedule.

Scheduling Method

As mentioned above, the objective of the RCPSP is to schedule the activities such that the makespan of the project is minimised (1), all the precedence relations are satisfied (2) and resource availability limits are not exceeded (3). The unknown variable of the problem is a vector of activity start times $X = \{t_n\}_{n=1}^N$. The generalised Workspace and Workflow Constrained Project Scheduling Problem (WWCPSP) can be introduced as a project makespan minimization problem (1), (2), (3) with the additional constraints on workspaces and workflows (6), (7). It is worth noting that the solution of the generalised problem includes not only activity start times, but also resource reallocation routes which would enable the activities to start on the scheduled times. Thus, the unknown variable of the WWCPSP problem is a structure $X = \{t_n, \{\bar{u}_{nk}\}_{k=1}^K\}_{n=1}^N$. Generalizing the RCPSP, the WWCPSP remains to be a NP-hard problem and requires long computation time even for finding suboptimal solutions. Let discuss the proposed scheduling method for the WWCPSP. The method adopts so-called serial scheduling scheme and provides for additional computational routines and heuristic rules to generate feasible schedules satisfying all the imposed requirements.

Scheduling Scheme

To resolve the WWCPSP problem the serial scheduling scheme mentioned in many works (Kolisch, 1996b) was adopted and advanced. It assumes a stage-wise algorithm extending a partial schedule (i.e. a schedule where only a subset of the activities has been scheduled and assigned a start time). Two disjoint activity-sets are associated with each stage, namely: the scheduled set S and the decision set D . The set S is formed by indices of the activities which were already scheduled and thus belong to the partial schedule. The decision set D contains indices of the unscheduled activities with every predecessor being in the scheduled set. In each stage one activity from the decision set is selected with a priority rule (in case of ties the next priority rule or the smallest activity number is applied to select the activity) and scheduled at its earliest precedence, resource and workspace feasible start time. Afterwards, the selected activity is removed from the decision set and put into the scheduled set. This, in turn, may place a number of activities into the decision set, since all their predecessors are now scheduled. The algorithm terminates at the stage number $j = N$, when all activities are in the scheduled set. The advanced serial scheme can be formally specified as follows:

INITIALISATION: $j := 1$ $S_j := \emptyset$;

WHILE $j < N$

BEGIN

COMPUTE

$$D_j := \{n = 1, \dots, N \mid n \notin S_j, \forall m = 1, \dots, M, n = Sc(m) \rightarrow Pr(m) \in S_j\}$$

$$n^* := \min_{n \in D_j} \{n \mid \theta(n) = \min_{n \in D_j} (\theta(n))\}$$

FOR $k = 1, \dots, K$

BEGIN

$$\bar{u}_{n^*k} := \{\bar{u} \mid \tau(\bar{u}) = \min_{\bar{u} \in \bar{U}_{n^*k}^{S_j}} (\tau(\bar{u}))\}$$

CREATE LINKS BY \bar{u}_{n^*k}

END

$$t_{n^*} := \text{earliest start preserving (2),(3),(6),(7) for } \forall t \mid t_{n^*} \leq t \leq t_{n^*} + d_{n^*}$$

$$D_{j+1} := D_j \setminus n^*$$

$$S_{j+1} := S_j \cup n^*$$

$$j := j + 1$$

END;

STOP;

Within the presented scheme at every step $j = 1, \dots, N$ a decision set D_j is formed and updated according to precedence relations. Using a priority rule function $\theta(n)$, the priority values are computed for all activities from the decision set and the activity with the maximum priority value is selected. Different priority rules are admitted within this scheme.

For the prioritised activity a_{n^*} and for each its consumable resource r_k an optimum route \bar{u}_{n^*k} is determined to minimise the transfer time $\tau(\bar{u}_{n^*k})$ at the set of all the possible routes $\bar{U}_{n^*k}^{S_j} = \left\{ \left\{ \Delta u_{nk}^{n_m} \right\}_{m=1}^{M(n,k)} \mid n_m \in S_j, m = 1, \dots, M(n,k) \right\}$, originating from the workspaces of the scheduled activities whose indices are already contained in the set S_j :

$$\bar{u}_{n^*k} := \{\bar{u} \mid \tau(\bar{u}) = \min_{\bar{u} \in \bar{U}_{n^*k}^{S_j}} (\tau(\bar{u}))\} \quad (8)$$

Having got the resource route, the transfer links are created with the lags corresponding to the time delays $\tau^k(\Delta u_{n^*k}^{n_m}, i(n_m, k), i(n^*, k))$ for each portion of the delivered resource $\Delta u_{n^*k}^{n_m}$. Finally, the prioritised activity a_{n^*} is scheduled so that both precedence relations (2), resource limits (3), workspace mitigation constraints (6), and induced workflow links conditions (7) are satisfied. The decision and scheduled sets are updated properly and the method proceeds to the next step.

Priority Scheduling Rules

The number of priority rules proposed is relatively high. The MTS, LST, GRPW, WRUP, LFF, MSLK rules mentioned above well suit to the RCPSP problems, but fully ignoring the spatial factors unlikely they would remain workable for the considered WWCPSP statements. Once the workspace and workflow constraints have been specified, effective priority rules for the WWCPSP problem can be proposed. In order to prevent the workflow disturbance and to keep resource traffic reasonable, the rule should minimise the resource moving time. It can be reached if the resources are supplied from the nearest workspaces with a minimal transfer time according to the function (8). If the units placed in a nearby workspace are not enough for the scheduled activity a_n , then the search is propagated over distant workspaces with an expectation that the requested units can be collected from several workspaces. The transfer time can be estimated for every sort of the requested resource and for every reallocated portion. This time may have impact to earliest start of the scheduled activity. As a result, the activity may be delayed by Δd_n due to all the resource transfers. The activities having minimum delay with respect to the original duration $\Delta d_n/d_n$ should be prioritised to avoid high resource traffic. We call this priority rule by a Moving Delay Ratio (MDR) and apply it as the first priority rule invoked by the function $\theta(n)$. In case of ties, LFF or MSLK are applied as secondary rules for the function $\theta(n)$.

Resource Reallocation Model

First of all, the method is based on an assumption that the time transfer function can be simplified and its dependency on the amount of reallocated resource units can be represented by a separate multiplier so that

$$\tau^k(\Delta u, i', i'') = \frac{1}{s^k(\Delta u)} \cdot \rho^k(i', i''), \quad (9)$$

where the value $s^k(\Delta u)$ plays role of the speed of the resource transfer and the factor $\rho^k(i', i'')$ means the length of a traversing path from the workspace $w_{i'}$ to the workspace $w_{i''}$. The second factor can be given in a tabular form $\{\rho_{i',i''}^k\}$ providing the path lengths among all the workspace pairs. Unfortunately, the use of the tabular form looks unrealistic for large projects as it would require manual input of huge data. An analytical form based on Euclidean, Manhattan or maximum norm for the 3-dimensional vector connecting geometric centers or corners of the workspaces is more convenient for practical purposes. However, it may produce wrong estimates of the path lengths and prevent right choice of next activities when traversing the project space. As an example, the distance from one room to an adjacent room along a corridor may be the same as the distance from the room to an upper room located at the next floor. But the path lengths between the workspaces seem to be essentially different. Moreover, if the activities are prioritised so that the resources to move in nearest, quickly reachable workspaces, norm-based estimates may become error-prone.

A more promising way is to use so-called space-filling curves and to define the spatial factor by means of a distance function on these curves. Figure 3 presents row-wise (a), prime row-wise (b), spiral-wise (f), U-wise (i) space-filling curves as well as the curves based on the well-known orderings by Morton (c), Peano-Hilbert (d), Cantor (e) and Gray (g, h). Every space-filling pattern can be generalised for the 3-dimensional case and adapted to the project space by means of altered orientations of axes of the underlying coordinate system, thereby producing 48 particular curves. Figure 4 illustrates this variety by giving a few particular curves for 3-dimensional row-wise and Peano-Hilbert space-filling patterns. By choosing one of the patterns, adjusting its orientation in the project space and setting the cell

sizes along different dimensions, the user defines a simple automatic routine for estimating resource reallocation time. If the chosen pattern and made adjustments match to a real project environment, the estimates become realistic so they can be applied when deciding on activity priorities and keeping workflows continuous.

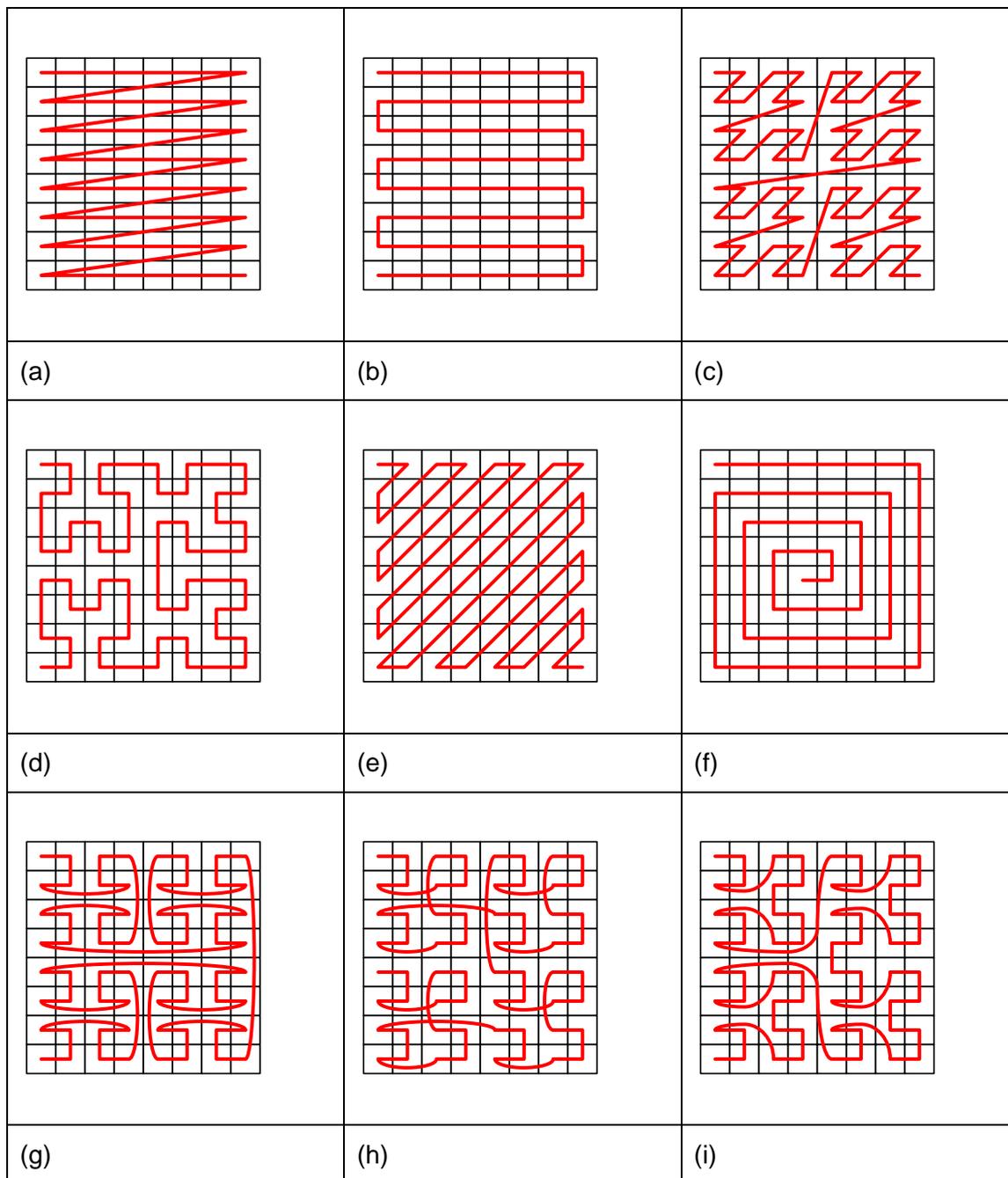


Figure 3: Space-filling curves for the transfer time estimation.

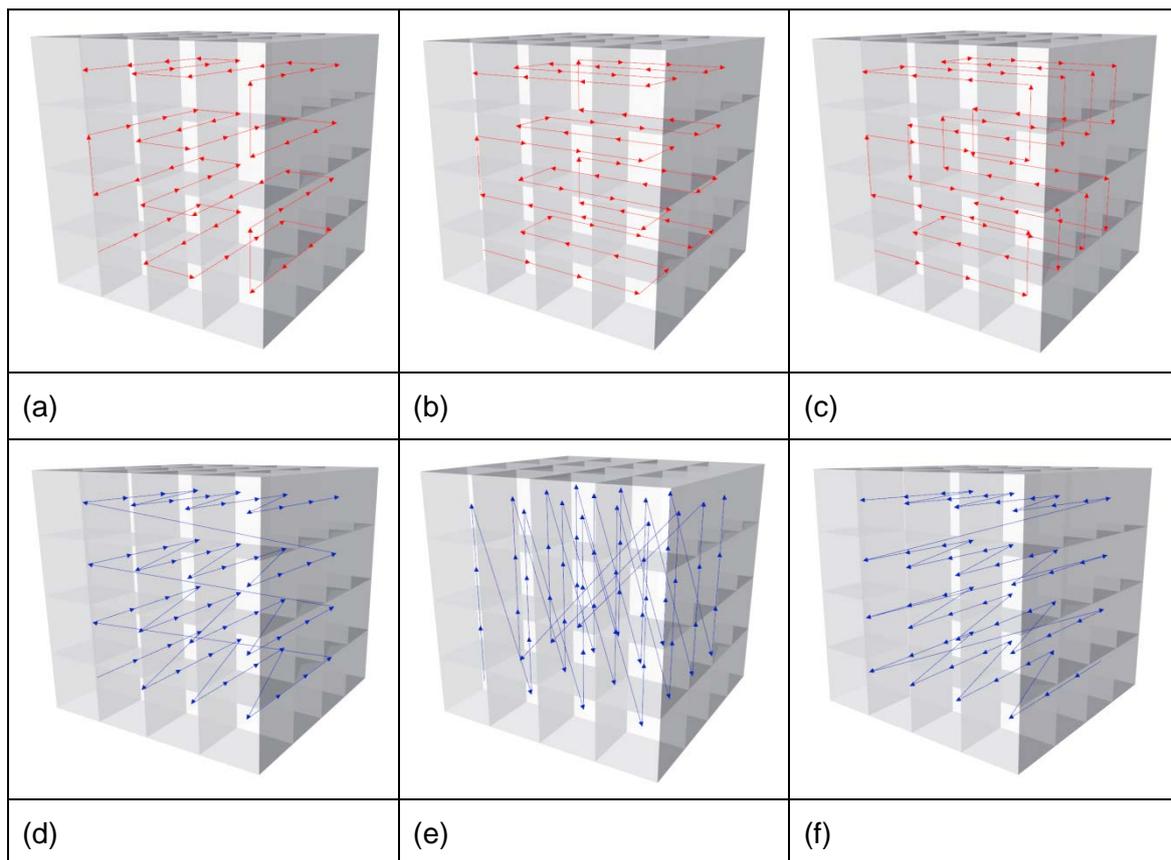


Figure 4: Space-filling curves generalised for 3D case.

Computational Experiments

A simple project on refurbishing a small hotel with partial replacement of supply lines has been used for computational experiments. The hotel is a two-storeyed building having 34 rooms of three types varied in size: standard, studio and family. Time required to refurbish a room depends on its size and is 2 days for standard, 4 days for studio and 6 days for family room. Refurbishing is performed by a crew of workers room-by-room sequentially. The sequence of rooms selected for refurbishing has no matter. Replacement of supply lines is accomplished by another crew in parallel with refurbishing and subdivided into 5 stages, each of 10 days. The first stage includes works related to all the hotel building, the other 4 stages can be performed after the first one has been finished and cover only one building aisle (left and right at the first and second storey correspondingly).

Traditional planning systems usually consider workspace as a spatial resource and assign it to all the activities that should be performed in it to control its availability. As applied to the considered project, 34 spatial resources (each represents a separate room) are created. All the resources are assigned to the activity representing the first stage of supply line replacement. Resources representing rooms located at the corresponding hotel aisle are assigned to other supply line replacement activities. Only one spatial resource is assigned to the activity on refurbishing the corresponding room. Then the activities are scheduled according to a predefined priority rule, for example, latest finish time (LFT). Scheduling algorithm based on this rule is simple enough: activities with the largest duration (or latest finish time) should be scheduled first. An activity with the smallest index is selected first for

scheduling among ones with equal duration. The prepared schedule is presented in the Figure 5a.

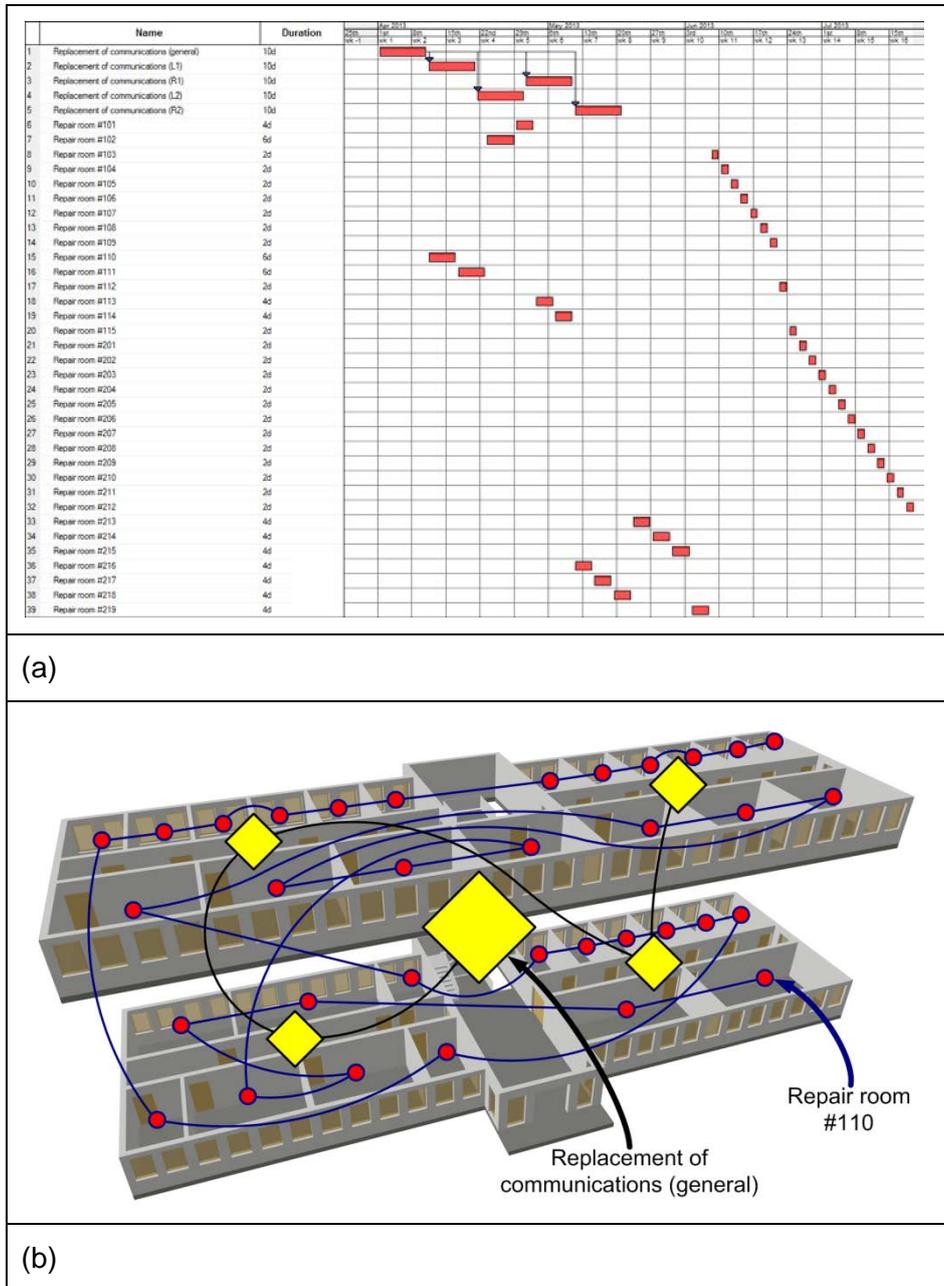


Figure 5: Hotel refurbishing schedule prepared using the traditional RCPSP method.

Visualisation of crew movement routes (see Figure 5b) demonstrates obvious shortcoming of the prepared schedule. According to it the refurbishing crew is forced to move 5 times from one aisle to another at the second storey, 3 times — from one storey to another, 3 times — from one aisle to another at the first storey. The supply line replacement crew moves 3 times from one storey to another. Such chaotic movement is inconvenient for the workers and may lead implicitly to overheads.

Conceptual difference of the proposed algorithm consists in taking into account spatial factor during scheduling procedure. In addition, the model of workspaces used in it is more accurate and flexible than reducing the workspaces to resources utilised in classical approaches. It allows loading the workspaces as much as possible. Being applied to the discussed project, it takes into account the fact that two crews can share the same workspace at the same time if allowable by its utilisation factor. One more difference of the algorithm is that the prioritization of activities depends on resource transfer time: activities with the minimum transfer time should be performed first. The time is estimated when bypassing possible routes of resource reallocation between workspaces. Activities with equal priority are scheduled according to classical approaches. The prepared schedule is presented in Figure 6a. It is essential that the project makespan according to the schedule prepared using the proposed method is 10 days smaller than in the previous example.

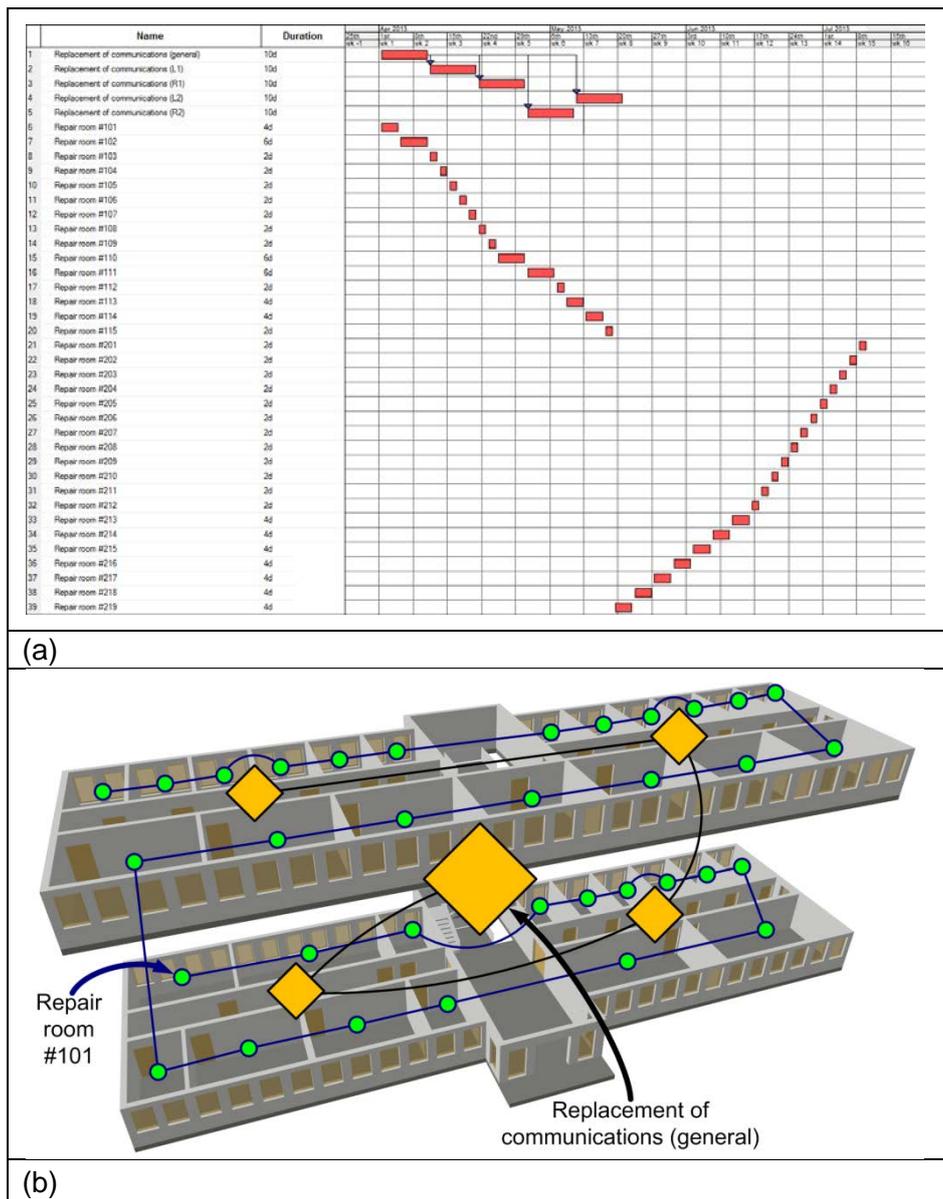


Figure 6: Hotel refurbishing schedule prepared using the WWCPSP method.

Figure 6b demonstrates that the crews should move from one storey to another only once. This is more reasonable and convenient comparing to the previous schedule. Thus, the proposed scheduling algorithm and the corresponding priority rule does not increase the total project duration and improve quality of workflow comparing to the classical approaches.

Conclusions

Thus, the new WWCPSP problem generalizing the classical RCPSP by taking into account spatial factor has been stated and formalized. The effective scheduling method for the problem has been proposed, investigated and approved. Like the classical approaches it tends to minimise the project makespan while satisfying timing constraint, precedence relations and not exceeding resource utilisation limits. In addition, the method takes into account workspace congestion and workflow disturbance factors and allows not only to determine availability of resources to perform the project activities, but also to control overloading of workspaces and to minimise time overheads required for reallocation of resources. The conducted computational experiments showed that the method generates feasible schedules near to optimal solution at least for the low-scale benchmark problems. The reached advantages allow employing the method for the industrial needs. Such activities are planned for the next research phase.

References

- Ahuja H. N. 1976, *Construction Performance Control by Networks*, John Wiley & Sons, New York.
- Boctor F. F. 1990, 'Some efficient multi-heuristic procedures for resource constrained project scheduling', *European Journal of Operational Research*, 49, 3-13.
- Bowers J. A. 1995, 'Criticality in resource-constrained networks', *Journal of the Operational Research Society*, 46(1), 80-91.
- Chavada R., Dawood N & Kassem M. 2012, 'Construction workspace management: the development and application of a novel nD planning approach and tool', *Journal of Information Technology in Construction*, 17, 213-36.
- Choo H. J. & Tommelein I. D. 1999, Space Scheduling Using Flow Analysis, *Proceedings IGLC-7*, 299-312.
- Chua D., Yeoh K & Song Y. 2010, 'Quantification of spatial temporal congestion in Four-dimensional computer-aided design', *Journal of Construction Engineering and Management*, 136 (6), 641-9.
- David E. W & Patterson J. H. 1975, 'A comparison of heuristic and optimum solutions in resource-constrained project scheduling', *Management Science*, 21(8), 944-55.
- Ellips M & Davoud S. 2007, Classic and Heuristic Approaches in Robot Motion Planning - A Chronological Review, *Proceedings of world academy of science, engineering and technology* 23, 101- 6.
- Hegazy T. 1999, 'Optimization of resource allocation and leveling using genetic algorithms', *Journal of Construction Engineering and Management*, 125(3), 167-75.
- Kolisch R., Sprecher A & Drexl A. 1995, 'Characterization and generation of a general class of resource-constrained project scheduling problems', *Management Science*, 41(10), 1693-703.
- Kolisch R. 1996a, 'Efficient priority rules for the resource-constrained project scheduling problem', *Journal of Operations Management*, 14, 179-92.
- Kolisch R. 1996b, 'Serial and parallel resource-constrained project scheduling methods revisited: Theory and computation', *Journal of Operations Management*, 90, 320-33.
- Lavalle S. M. 2006, *Planning algorithms*. Cambridge University Press, UK.

Semenov V. A., Kazakov K. A & Zolotov V.A. 2011, Virtual Construction: 4D Planning and Validation, *Proceedings of the XI International Conference on Construction Applications of Virtual Reality*, 135-42.

Pai S. K., Verguese, P & Rai, S. 2013, 'Application of Line of Balance Scheduling Technique (LOBST) for a Real estate sector', *International Journal of Science, Engineering and Technology Research (IJSETR)*, 2(1), 82-95.

Thabet W. Y. & Beliveau Y. J. 1994a, 'Modeling Work Space to Schedule Repetitive Floors in Multistory Buildings', *Journal of Construction Engineering and Management*, 120(1), 96-116.

Thabet W. Y & Beliveau Y. J. 1994b, 'HVLS: Horizontal and Vertical Logic Scheduling for Multistory Projects', *Journal of Construction Engineering and Management*, 120(4), 875-92.

Yeoh K. W & Chua D.K.H. 2012, 'Mitigating Workspace Congestion: A Genetic Algorithm Approach', *EPPM 2012 Conference*, 107-18.

Zouein P. P & Tommelein I. D. 1999, 'Dynamic Layout Planning Using a Hybrid Incremental Solution Method', *Journal of Construction Engineering and Management*, 125(6), 400-8.