PTIMIZATIONOFRESOURCEALLOCATIONANDLEVELINGUSINGGENETICALGORITHMS

Syed Fazil, Rafey Ahmed, Sri Charan
Department of civil
fazil001@gmail.com,rafay12385@gmail.com, charansri0123@gmail.com
ISL Engineering College.
International Airport Road, Bandlaguda, Chandrayangutta Hyderabad - 500005 Telangana, India.

ABSTRACT: Resource allocation and leveling are among the top challenges in project management. Due to thecomplexity of projects, resource allocation and leveling have been dealt with as two distinct subproblems solvedmainly using heuristic procedures that cannot guarantee optimum solutions. In this paper, improvements areproposed to resource allocation and leveling heuristics, and the Genetic Algorithms (GAs) technique is used tosearch for near-optimum solution, considering both aspects simultaneously. In the improved heuristics, randomprioritiesareintroducedintoselectedtasksandtheirimpactonthescheduleismonitored.TheGAprocedurethen searches for an optimum set of tasks' priorities that produces shorter project duration and better-leveledresource profiles. One major advantage of the procedure is its simple applicability within commercial projectmanagement software systems to improve their performance. With a widely used system as an example, a macroprogram is written to automate the GA procedure. A case study is presented and several experiments conductedtodemonstratethemultiobjectivebenefitoftheprocedure andoutlinefutureextensions.

INTRODUCTION

Few companies can remain competitive in today's highly competitive business environment without effectively manag- ing the cost of resources. In practice, basic PERT and CPM scheduling techniques have proven to be helpful only when the project deadline is not fixed and the resources are not con- strained by either availability or time. Since this is not prac- tical even for small-sized projects, several techniques have been used to modify CPM results in account of practical con- siderations. In dealing with project resources, two main types of techniques have been used: resource allocation and resource leveling. Resource allocation (sometimes referred to as con- strained-resource scheduling) attempts to reschedule the proj- ect tasks so that a limited number of resources can be effi- ciently utilized while keeping the unavoidable extension of the project to a minimum. Resource leveling (often referred to as resource smoothing), on the other hand, attempts to reduce the sharp variations among the peaks and valleys in the resource demand histogram while maintaining the original project du- ration (Moselhi and Lorterapong 1993). These techniques, as such, deal with two distinct subproblems that can only be ap- plied to a project one after the other rather than simultane- ously. Accordingly, they do not guarantee (either individually or combined) a project schedule that minimizes the overall project time or cost (Karshenas and Haber 1990).

In this paper, an attempt is made to develop a practical procedure that searches for a near-optimum solution to re- source allocation and leveling, simultaneously. The paper starts with a brief description of the advantages and limitations of current optimization-based and heuristic approaches. Indi- vidual improvements to existing heuristics are then proposed and tested on a case study. A multiobjective optimization using the genetic algorithms (GA) technique is then described and coded in a macro program. The performance of the proposed GA procedure is then evaluated on the case study, and rec- ommendations made.

RESOURCE ALLOCATION AND LEVELINGHEURISTICS

Limited-resource allocation algorithms deal with a difficultproblem thatmathematiciansrefertoasa‘‘largecombinatorial problem.’’Theobjectiveistofindthescheduledurationthatis shortest, as well as consistent with specified resource limits.There exist optimization methods as well as heuristic methodsforsolving the resource allocation problem that go back intime to the 1960s (e.g., Wiest 1964). Various approaches havebeen formulated to solve the problem optimally, including In-tegerProgramming,branch-and-bound,andDynamicPro-gramming (Gavish and Pirkul 1991). None of these, however,iscomputationallytractableforanyreal-lifeproblemsize,rendering them impractical (Moselhi and Lorterapong 1993;Allam1988).

Alternatively, heuristic approaches have been proposed forsolving the resource allocation problem. These approaches ap-ply selected heuristic (rules) that are based on activity char- acteristics, such as the ‘‘minimum total-slack’’ rule, to prior- itizetheactivitiesthatcompeteforthelimitedresource.Accordingly, the resource is given to the top-ranked activitiesand the others are delayed. When ties occur during the imple-mentation of a rule (e.g., when two or more activities have thesame total slack), another rule such as ‘‘shortest duration’’ canbe used to break the tie. The scheduling process, as such, startsfrom the project's start time, identifying eligible activities ac-cording to the network logic and resolving the over-require- mentsofresources using the selected set of heuristic rules.Theprocess,assuch,ensuresthatallproject activities arescheduled without violating the logical relationships or the re-source constraints. However, this comes on the expense of thetotal project duration, which often exceeds the duration deter-minedbytheoriginalCPManalysis.

Heuristic rules have the advantage of being simple to un- derstand, easy to apply, and very inexpensive to use in com- puter programs. They are able to rationalize the schedulingprocessandmakeitmanageableforpractical-size projects(Talbot and Patterson 1979). Furthermore, research has iden-tified rules such as the ‘‘least total-slack’’ and the ‘‘earliestlate-start,’’ which generally provide good solutions (Davis andPatterson 1975). Almost all commercial software for planningand scheduling, therefore, utilizes heuristic rules to provideresource allocation capabilities. Despite these benefits, how-ever, heuristic rules perform with varying effectiveness whenused on different networks, and there are no hard guidelinesthat help in selecting the best heuristic rule to use for a givennetwork. They, as such, cannot guarantee optimum solutions.Furthermore, their drawbacks have contributed to large incon-sistencies among the resource-constrained capabilities of com-mercial project management software, as reported in recentsurveys(HegazyandEl-Zamzamy1998;Johnson1992).

Resource-leveling algorithms, on the other hand, attempt toreducepeakrequirementsandsmoothoutperiod-to-period fluctuations in resource assignment without changing projectduration. Typical resources considered include a rented pieceof equipment that needs to be returned early or a number ofskilled workers who need to be hired for the job. Optimalsolutionsfor theresource-levelingproblemarebasedonmixedinteger program formulations (Shah et al. 1993; Easa 1989).Such formulations are NP-complete and optimal solutions arereached for small-sized construction projects only. Heuristicalgorithmsarethereforeneeded.

A well-known heuristic algorithm is the minimum momentalgorithm (Harris 1978). The objective in this algorithm is tominimize daily fluctuations in resource use while keeping thetotal project duration unchanged. As a proxy to this objective,the algorithm minimizes the moment of the resource histogramaroundthehorizontalaxis(time,calculationspresentedlaterin more detail). To accomplish this objective, the algorithmstarts from an early start schedule and shifts noncritical activ-itieswithintheirfloattimessoastocausenoprojectdelay.At each time step, the shift(s) that yields the maximum reduc-tion in the histogram moment is selected. Despite the simplenature of resource-leveling heuristics and their wide imple-mentation on commercial project management software, theycan only produce good feasible solutions and by no meansguaranteeanoptimumsolution.

IMPROVING RESOURCE-ALLOCATION HEURISTICSUSINGBIASEDPRIORITIES

Since it is not possible to select an optimum heuristic ruleforagivenprojectnetwork,onecommonprocedureistotrya series of heuristic rules and then select the schedule withminimum duration. This procedure, however, has little diver-sity since the number of effective rules to enumerate is smallanditisnotexpected thatlesseffectiveruleswillchangemuchwhen effective rules are not improving the schedule. There-fore, without introducing new rules or changing the mechanicsof heuristic procedures, a simple approach of forcing randomactivity priorities is presented to improve the goodness of theschedule. The concept is demonstrated on a case study of aproject withtwentyactivities and six resources.Thecasestudydata including activities' resource requirements and daily re-sourcelimitsispresentedinTable1.Thisdatawasinputto Microsoft (MS) Project software (Microsoft Project 1995) forquickanalysis.

Without considering the given resource constraints, the totalproject duration, determined by simple CPM analysis, is 32days. When the resource-leveling feature (leveling is used inthe software's terminology for both allocation and leveling) ofMS Project was set to ''Automatic,'' total project duration wasextended to 49 days, avoiding resource over-allocations. Thissolution was obtained using the software's ''standard'' set ofheuristic rules, which maintains logical relationships and ap-plies the ''minimum total slack'' rule to resolve conflicts. Thesame results were also obtained using the ''minimum totalslack''ruleonPrimaveraSoftware(Primavera1995)asahigh-end system. Several other heuristic rules were also triedonPrimaverasoftware,withoutimprovingtheschedule.

Aprojectdurationof49daysis,therefore,thebestresultthatcan be obtained from widely used commercial software. It isnotedthatthisresultisobtainedwhenallprojectactivitieshavethesamepriorityleve.

Most commercial scheduling software systems allow usersto specify priority levels to activities. MS Project implementsthat in a direct manner by allowing users to select among eightpriority levels (''Highest,'' ''High,'' etc., to ''Lowest''), andassign it in a simple spreadsheet form. The software also pro-vides a second set of heuristic rules for resource allocation inwhich activity priority takes precedence over its ''standard''setofheuristicrules.Itispossible, therefore, to introducesome bias into some activities and consequently monitor theimpactontheschedule. Asanexample,considerthe casewhenonlyactivity(R)inthepresentcasestudyisgiven''Highest'' priority while all others are set to ''Lowest.'' Withthis limited change to the original schedule, the project dura-tion substantially decreased to 46 days (Fig. 1), one of thesolutions for that particular example obtained by Talbot andPatterson (1979) using optimization. This simple approach istherefore proven to provide better results than existing heuris-tics.

Since it is not possible to readily identify, from a givennetwork, which activities to assign higher priorities than othersto improve the schedule, a simple iterative procedure may beused.AflowchartofsuchaprocedureispresentedinFig.2.It starts by initializing the scheduling software by setting itsresource allocation feature to ''Automatic'' and defining a setof heuristic rules, ''activity priority'' being the leading one.Afterwards,each activity in the project is selected in turn,given ''highest'' priority over all others, and the consequentproject duration is monitored. If the project duration decreasesat any step in the process, corresponding activity priorities aresaved and the process continues to improve the schedule fur-ther.Itisalsopossibletoautomatethisprocedurebywritinga simple macro on the scheduling software. Despite its per-ceived benefit, however, the main shortcoming of this proce-dure is its inability to identify an optimum set of activities'priorities that reduces project duration the most. This issue isdealtwithlaterusingtheGA.

IMPROVINGRESOURCELEVELINGHEURISTICSUSINGDOUBLEMOMENTS

In the course of optimizing resource allocation, the schedulerepeatedly changes and along with it are the daily demands ofresources. It is the objective of project managers, therefore, tooptimize both the allocation and the leveling aspects of re-sources. As mentioned previously, the minimum moment al-gorithm has been used as a heuristic approach to calculate ameasure of the fluctuations in daily resource demands. This isrepresented in Fig. 3(a), where Histogram 1 and Histogram 2are two alternative resource histograms, both having a totalareaof40resourcedays(i.e.,equaltotalresourcedemands).

MULTIOBJECTIVEOPTIMIZATION SEARCHUSINGGENETICALGORITHMS

Individualoptimizationof resource allocation or levelinghas not been a simple task, let alone their simultaneous opti-mization.Giventhemodifiedheuristicspresentedinthispaper,the objective can be restated, in a heuristic sense, as the searchfor a near-optimum set of activities' priorities that minimizesthe total project duration under resource constraints while alsominimizing the appropriate moment(s) of selected resources.This objective has a direct relationship to project cost mini-mization, which cannot be adequately achieved using mathe-maticaloptimizationtechniques. A schedule that efficientlyemploys limited resources, avoids daily fluctuation, and re-duces project duration is eventually less costly. To deal withthesemultiobjectives,asearchtechniquebasedonartificial intelligence, GAs, is used. Analogous to natural selection andgenetics in reproduction, GAs have been successfully adoptedto solve many science and engineering problems (Feng et al.1997; Hegazy and Moselhi 1994). GAs also have been provento be an efficient means for searching optimal solutions in alargeproblemdomainsuchastheoneathand.

GAsare,inessence,optimizationsearchproceduresin-spiredbythebiologicalsystems'improvedfitness throughevolution. GAs employ a random-yet-directed search for lo-cating the globally optimal solution. Typically, GAs require arepresentation scheme to encode feasible solutions to the op-timization problem. Usually this is done in the form of a stringcalledachromosome(orgene).Eachgenerepresentsonemem ber, i.e., one solution, that is better or worse than othermembers in a population. The fitness of each gene is deter-mined by evaluating its performance with respect to an objec-tive function. To simulate the natural ''survival of the fittest''process, best genes exchange information to produce offspringthat are evaluated in turn and can be retained only if they aremore fit than the others in the population. Usually the processis continued for a large number of offspring generations untilanoptimumgeneisarrivedat.

Implementing the GA technique for the problem at handinvolved five primary steps: (1) Setting the gene structure; (2)deciding the gene evaluation criteria (objective function); (3)generating an initial population of genes; (4) selecting an off-spring generation mechanism; and (5) coding the procedure ina computer program. First, the gene structure was set as astring of elements, each corresponding to a priority level as-signed to an activity, as shown in Fig. 4. As such, each generepresents one possible solution to the problem. To evaluategenes,anobjectivefunctioncan be constructed by elicitingthe user's preference (or weights) among the multiobjectives.For example, assume a project with (r) resources, initial proj-ect duration D_0determined by any resource allocation heuristicrule, initial $M_x$moment of every (j) resource (M_{xj0}), and initial$M_y$moment of every (j) resource (M_{yj0}). The values D_0, M_{xj0}'s,and M_{yj0}'s are therefore constants associated with the best so-lution provided by the scheduling software, before the GAprocedure is applied. The user then needs to input the weight$W_d$of his preference in minimizing project duration and theweightsW_j'sofhispreferenceinlevelingeveryresource (j).In addition, the user needs to input the type of leveling mo-ment(i.e.,M_x,M_y,orM_x+M_y)thatneedstobeminimizedfor every resource (j). The weights and moment types are alsoconstantsrepresentingtheprojectmanager'sobjective.

When a gene (i) is being evaluated, its priority values areassignedtotheprojectactivitiestoproduceanewschedule

Once the gene structure and fitness function are set, GA'sevolutionary optimization takes place on a population of par-ent genes. The simplest way to generate that population israndomly, if no information is available on any activity thatmust have a fixed priority level. Population size (number ofgenes)isalsoanimportantfactoraffectingthesolutionandtheproc essingtimeitconsumes. Larger population size (onthe order of hundreds) increases the likelihood of obtaining aglobal optimum; however, it substantially increases processingtime. In the present application the user is given the flexibilityto input the population size. Once the population is generated,thefitnessofeachgeneinthispopulationisevaluatedusin gthe objective function (5), and accordingly its relative merit iscalculatedasthegene'sfitnessdividedbythetotalfitnessofallgene s.

The reproduction process among the population memberstakes place by either crossover or mutation, resembling

naturalevolution. Crossover (marriage) is by far a more common pro-cess and can be conducted by selecting two parent genes, ex-changing their information, and producing an offspring. Eachof the two parent genes is randomly selected in a manner suchthat its probability of being selected is proportional to its rel-ative merit. This ensures that the best genes have a higherlikelihood of being selected, without violating the diversity ofthe random process. Also, the exchange of information be-tween the two parent genes is done through a random process(Fig. 5). As opposed to crossover, which resembles the mainnaturalmethod of reproduction (Goldberg 1989), mutation isa rare process that resembles the process of the sudden gen-eration of an odd offspring that turns to be a genius. This canbe done by randomly selecting one gene from the populationand then arbitrarilychangingsomeofitsinformation.Theben-efit of the mutation process is that it can break any stagnationintheevolutionaryprocess,avoidinglocalminimums.
Once an offspring is generated by either method, it is eval-uatedinturnandcanberetainedonlyifitsfitnessishigher

PROCEDURE AUTOMATION AND EXAMPLEAPPLICATION

Implementing the proposed GA procedure on commercialscheduling software simplifies the implementation process andprovides project managers with an automated tool to improvethe results of their familiar software. In this study MicrosoftProject software is selected for implementing the GA proce-dure,for the reasons mentioned earlier as well as its ease ofuse and programmability features. The detailed GA procedureis outlined in Fig. 6. Using the macro language of MicrosoftProject,theprocedurewascodedandthenusedtosearchfo ranoptimumscheduleforthecasestudyathand.
Forsimplicity,onlyoneresource(R4)ofthesixresourcesin the present case study is assumed to be critical. As discussedpreviously, the software's initial solution to the resource-con-strained schedule used ‘‘Lowest’’ priority for the project's 20activities (column 2 of Table 2), producing a schedule of 49days, in addition to $M_x$of 2,409, and $M_y$of 7,231 for resourceR4. The GA optimization-search procedure was used to con-duct four experiments with different objectives, as outlined inthe second and third rows of Table 2. After initial experimen-tationwithdifferentpopulationsizesandnumberofoff-springs, a population of 200 genes and offspring of 1,000 wasfound to be a reasonable compromise between diversity andprocessing time for this size of problem. Accordingly, thesewerefixedforallexperiments.Also,toavoidstagnation,crosso ver operation was set to be responsible for 95% of off-spring generations while mutations was set to only 5%. Oncethe procedure was activated, an input screen, shown in Fig. 7for Experiment 2, was displayed, requesting user input regard-ing GA parameters and the weights needed to formulate theobjective function. The GA procedure then performed the op-timizationsearch,producinganoutputscreenasshownin Fig. 8.Theactivityprioritiesresultingfromthefourexperiments are shown in Table 2 along with the associated project dura-tionsandmomentcalculations.
It can be seen from the results of Table 2 that each exper-imentimprovedthescheduleinamannerthatisconsistentwith its objective. In Experiment 1 the objective was to solelyminimize project duration, and accordingly a 44-day schedulewas obtained (Fig. 9). This is 5 days shorter than the initialscheduleandisalso2dayslessthanthe46-dayscheduleofthe iterative process discussed earlier. Giving a 50% weight tominimizing the $M_x$of R4, Experiment 2 produced the smallestresource fluctuation moment of all the experiments (2,265).This was also reflected on the daily fluctuation range of R4demand, which decreased from the initial 12 units to 10 units(Table 2). This experiment also decreased project duration to45 days. Experiment 3 attempted to equally minimize the re-source utilization moment (M_y) and the project duration, re-sulting in best improvements to both. Experiment 4 also at-tempted to minimize the three aspects of project duration, R4fluctuation, and R4 utilization period, resulting in improve-ments to the three. Based upon these results, the case studyclearly shows the benefits of the GA procedure in optimizingboth resource allocation and leveling to improve schedulingresults over those of existing heuristic procedures and com-mercial scheduling software systems. Clearly these benefits, intermsofshorterdurationandbetterresourceutilization,canbe readily translated into cost savings as a function of indirectcost,incentive gains, and reduced resource rental or salaryamounts.,forexample,canspecifya custom activity codecalled ‘‘priority,’’ containing a number that represents the pri-ority level of each activity. This code can then be used as theleading heuristic rule for resource allocation and for the im-plementationoftheGAprocedure.

SUMMARYANDCONCLUDINGREMARKS

Three main developments were made in this paper with re-spect to improving the resource management of projects: (1)An effective improvement to resource allocation heuristics us-ing random activity priorities; (2) a practical modification toresource leveling heuristics using a double-moment approach;and (3) a multiobjective optimization of both resource allo-cation and leveling using the genetic algorithms technique. Us-inga widely used project management software, a macro pro-gram was written to automate the GA procedure and a casestudy was used to demonstrate its benefits and future improve-mentsandextensions.
In recent years, project management software systems havebeen improving continuously and recent versions have exhib-ited better interfaces, integrated planning and control features,and Internet capabilities. Yet, basic project management func-tions such as resource allocation, resource leveling, and time-cost trade-off analysis have been the least improved. Still, tosome practitioners software systems provide merely powerfulpresentation capabilities and real savings can be achieved onlyby putting a hammer to a nail. It is hoped that practical im-plementations of new approaches such as genetic algorithmsjustify the effort spent in proper planning and scheduling askeys to effective project management and ultimately to actualsavingsinprojecttimeandcost.

REFERENCES

Allam, S. I. G. (1998). ‘‘Multi-project scheduling: A new categorizationfor heuristic scheduling rules in construction scheduling problems.’’*J.Constr.Mgmt.andEconomics*,E&FNSpon,6(2),93–115.Davis, E. W., and Patterson, J. H. (1975).‘‘A comparison of heuristic andoptimum solutions in resource-constrained project scheduling.’’*Mgmt.Sci.*,21(8),944–955.
Easa, S. (1989).‘‘Resource leveling in construction by optimization.’’*J.Constr.Engrg.andMgmt.*,ASCE,115(2),302–316.
Feng,C.,Liu,L.,andBurns,S.(1997).‘‘Usinggeneticalgorithmstos olve construction time-cost trade-off problems.’’ *J. Comp. Civ. Engrg.*,ASCE,11(3),184–189.
Gavish, B., and Pirkul, H. (1991).‘‘Algorithms for multi-

resource gen-eralizedassignmentproblem.'' *Mgmt.Sci.*,37(6),695–713.

Goldberg, D. E. (1989).*Genetic algorithms in search, optimization andmachinelearning*.Addison-Wesley,Reading,Mass.

Harris, R. (1978). *Resource and arrow networking techniques for con-struction*.Wiley,NewYork.

Hegazy, T., and El-Zamzamy, H. (1998).''Project management softwarethatmeetthechallenge.''*CostEngrg.J.*,4(5),25–33.

Hegazy, T., and Moselhi, O. (1994).''Analogy-based solution to markupestimationproblem.''*J.Comp.Civ.Engrg.*,ASCE,8(1),72–87.

Johnson,R.(1992).''Resourceconstrainedschedulingcapabilities ofcommercialproject management software.'' *Proj.Mgmt. J.*, 22(4),39–43.

Karshenas, S., and Haber, D. (1990). ''Economic optimization of con-struction project scheduling.''*J. Constr. Mgmt. and Economics*, E&FNSpon,8(2),135–146.

Li,H.,andLoing,P.(1997).''Usingimprovedgenetic algorithms tofacilitate time-cost optimization.'' *J. Constr. Engrg. and Mgmt.*, ASCE,123(3),233–237.

Microsoft Project. (1995). *Reference manual, ver. 4.1 for Windows 95*.MicrosoftCorporation,Redmond,Wash.

Moselhi, A., and Lorterapong, P. (1993).''Least impact algorithm forresourceallocation.''*Can.J.Civ.Engrg.*,CSCE,20(2),180–188.

Premavera.(1995). *Reference manual, ver. 1.0 for Windows*.Primavera,BalaCynwyd,Pa.

Shah, K., Farid, F., and Baugh, J. (1993).''Optimal resource levelingusing integer-linear programming.''*Proc., 5th Int. Conf. in Comp. inCiv.&Bldg.Engrg.*,ASCE,Reston,Va.,1,501–508.

Talbot, F., and Patterson, J. (1979). ''Optimal methods for schedulingprojectsunder resource constrains.'' *Proj.Mgmt. Quarterly*, Dec.,26–33.

Wiest, D. (1964). ''Some properties of schedules for large projects withlimitedresource.''*OperationsRes.*,12,395–416.