SIMULATION-OPTIMIZATION IN GEOGRAPHIC MAINTENANCE SCHEDULING

Javier Otamendi
Universidad Rey Juan Carlos, Campus Vicálvaro
Facultad de Ciencias Jurídicas y Sociales
Departamento Economía Aplicada I
Paseo Artilleros s/n
28032 Madrid, Spain
E-mail: franciscojavier.otamendi@urjc.es

KEYWORDS
Optimization, Scheduling, Simulation, Maintenance.

ABSTRACT

An application of a combination of scheduling theory, simulation modelling and combinatorial optimization is presented. The system under study deals with human resource and assets management to perform maintenance in large support networks. The paper includes the development of the framework of the software tool that will be embedded into the commercial maintenance package called INGRID. Following the modelling approach, a fully detailed explanation of the data requirements, the decision variables and the output variables is given. A detailed discussion on the genetic algorithm that is going to be used is also provided. The routines have been developed in spreadsheets so that there is an easy link with the rest of the software application.

INTRODUCTION

The human resource management software called INGRID is beginning to be used both nationally and internationally. Its objective, as depicted in its last name, Geographic Management of Assets and Maintenance, is the geographical management of maintenance resources. In particular, the software gives the opportunity to know in real time the status of the resources, both human and assets, as well as the demand of tasks to be performed. The software is focused in assigning the maintenance tasks to the resources, with applications ranging from public gardens to railways.

INGRID is prepared, at the same time, to generate a database in a daily basis in which information about all of the human resource teams and all of the required tasks is included in a geographic interface that might be exported into a web page, all without the external intervention of any additional software package.

The improvements in the software call for the help in the assignment of tasks as well as in the usage of the program. The first improvement path, currently under way, is the compatibility of the program with a Geographical Information System (GIS) which will allow for the automatic detection of the position of any maintenance resource. Therefore, when a new assignment of a task is to be made, the positions are conveniently updated without human intervention.

The second improvement path, which is the subject of this article, is the automation of the process of assigning tasks to maintenance teams and assets. That is, future versions of the software will include the appropriate optimization routines to assign tasks according to an explicit cost function to minimize. The user will only need to enter a new task and press and execution button that will automatically call the necessary, available resources.

The Spanish Ministry of Education and Science have funded this optimization project, through the program PROFIT 2005. The project has been called SORM – Sistemas de Optimización de Recursos de Mantenimiento (Systems to Optimize Resources for Maintenance).

For the development of the framework of this management tool to be inserted into INGRID, the decision has been for the usage of a modelling approach. Therefore, a model that rightfully resembles the current human decision process has been developed. For that reason, as a first step, the main decision variables that are controlled by the decision maker have been identified. Similarly, the main output criteria have been listed. Besides, the necessary data variables to execute the model have been identified. As the second step, the proper routine to generate, evaluate and select the best resources to schedule has been selected, trying to jointly obtain efficacy, or attainment of the optimum solution, and efficiency, or attainment of the solution in a reasonable time. Following, each one of the parts of the framework for this optimization part of the INGRID software is described in detailed, starting with the objectives of the study.

Proceedings 21st European Conference on Modelling and Simulation
Ivan Zelinka, Zuzana Oplatková, Alessandra Orsoni ©ECMS 2007
ISBN 978-0-9553018-2-7 / ISBN 978-0-9553018-3-4 (CD)

OBJECTIVES

The main objective of the SORM project is the development and updating of a set of algorithms that are capable of optimally scheduling the human resources and assets that are available to perform maintenance tasks in large support networks.

As secondary, but necessary, objectives for the proper use of the scheduling software, the following might be mentioned:

1. Development of the analysis and parameterization methodology to implement this type of optimization algorithms.

2. Adapt the optimization system to be an analysis procedure to evaluate the new organizational structures or tasks processes:
   - Capabilities of the task force (specialists vs. non-specialists)
   - Larger geographical areas per team
   - Subcontracting

In this first stage of the study, the main focus is in the development of a prototype following the modelling approach, with later stages focusing on the application.

THE MODELLING APPROACH

Lawrence Boland proposes in his book "The Foundations of Economic Method" (Boland 1982) the utilization of a methodology dependent on the problem in hand (Problem-Dependent Methodology). In fact, he proposes modelling as a means to represent reality, each model representing a different problem. This methodology, or method of analysis, might be renamed as the modelling approach.

Three stages might be distinguished when following this approach (Figure 1):

1. Definition of the problem

2. Modelling of the system

3. Experimenting with the model and decision making

Problem Definition

The first stage consists of the conceptualization of the real system under study. A system might be defined as “process … or any combination of machines, methods, personnel or other resources which transform a given input in an output that present one or more observable responses” (Montgomery 1984). It is clear that the decision variables **x,** the data **z**, or phenomena that interrelate within the system, and the output criteria **y** that want to be quantified as a result of the interactions within the system must be specified.

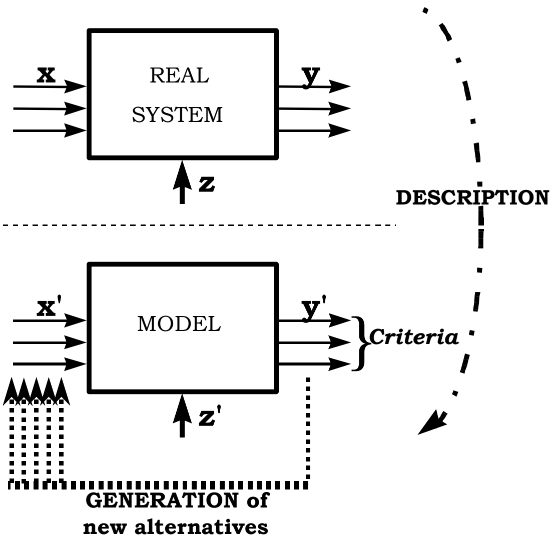

Figure 1: Description of the Real System into a Model

It seems essential that the problem is delimited, clearly specifying not only the variables and the data but also their interdependencies within the system. In very complex situations, it is also necessary to define the limits of the system and the appropriate level of detail according to the objectives of the study.

System Modelling

The study of a system is rarely performed directly on it. Either it is not possible to alter normal operations or it is not feasible to repetitively experiment with it so it is necessary to develop a model which acts as an abstraction of the real system and that is liable and ready for experimentation. The success of the analysis process, and consequently, of the implementation of the decision, clearly depends on this abstraction.

The model will be characterized by a set of input variables **x’** and input data **z’**, which are a representation of the corresponding of the real system, and another set of output variables **y’** that are a function of the input variables and the existing relationships between them.

Computer simulation has received a lot of attention in the last decades to abstract and model complex systems under uncertainty, in many areas but specifically in scheduling and sequencing (Fisher and Ittner 1999, Hershauer and Ebert 1975, Hollier 1968, Macaskill 1973, Moccellin and Nagano 1998, Wein and Chevalier 1992). The simulation models jointly consider efficacy, or degree of attainment of the optimal solution, and efficiency, or time to achieve it.

Experimentation With The Model And Decision Making

If the model is valid and credible, it is ready for experimentation and for decision making, that is, ready to search for the adequate values of the input variables.

Characterization of the criteria
The first stage of this third phase is the characterization of the criteria that allow for the evaluation of each alternative. The different criteria are computed out of one or several of the output variables **y’** of the model and out of the subjective information introduced by the decision maker.

Under certainty, the value of each criterion for each alternative is easily evaluated. However, two problems may arise. The first corresponds to the evaluation under uncertainty, since the output variables are not a single value anymore but a random variable. The second arises whenever there exist several criteria simultaneously, which might be aggregated so that the alternatives might be compared against each other.

Generation, Evaluation and Selection of Alternatives
In this second stage, the aim is to obtain information on the values of the input variables that improve the values of the output criteria.

Any method of analysis must generate, first, different alternatives or combinations of values for the parameters **x’**; then, it must evaluate the alternatives individually with respect to each of the criteria that characterize the system, and, finally, the algorithm must select the alternative that seems superior to the rest of alternatives.

For each possible combination of input values, an assignment of tasks in this article, it is necessary to evaluate the values for the different criteria. The number of alternatives is usually large enough to provoke the infeasibility of trying them all. Therefore, the combinatorial problem is usually not complex but very time consuming.

Several new methods have arisen in the last decades for problems in which the number of alternatives is very large. They have been named as combinatorial methods, or metaheuristics (April et al. 2003). Three different subgroups might be mentioned:

1. Simulated annealing (Eglese 1990), in which a new alternative is selected in the neighbourhood of the last simulated alternative.

2. Evolutionary algorithms like tabu search (Karaboga and Kalinli 1997), and genetic algorithms (Goldberg 1989). They search by building and evolving a subset of alternatives.

3. Metamodels, like neural networks (Hopfield and Tank 1985). They are used to algebraically represent the simulation model, facilitating then the optimization procedure.

All these methods have been reported to be very useful when the number of alternatives is very large, since they have good convergence properties. The solution of the scheduling problem in flexible manufacturing systems via genetic algorithms (for example, Fazlollahi and Vahidov 2001, Iyer and Saxena 2004, Yu and Liang 2001) is one of the examples most commonly mentioned. In this article, the scheduling process of maintenance tasks is the core of the analysis process.

THE SYSTEM

Operations

The system, in terms of its conception, is simple. When there is a need to perform a task, a maintenance task for this project, the computer software INGRID, using a task code as the lead, generates a list with the necessary resources to perform the task. There coexist two types of resources: human resources and assets (machines and machinery). The person who is in charge of the scheduling duties locates the available resources in their idle-state locations and assigns the task to a human team.

The team must pick up the necessary assets at their corresponding locations, go to the location where the task is to be performed and execute it. Once finished, the team should return the utilized resources back to their idle-state location and direct themselves to their initial point. There might be several tasks opened at the same time.

Decision variables

The human decision maker might have an influence in the result of the assignment process of maintenance tasks to teams. The decision implies that the resources just assigned are freezed and cannot be assigned to any other task. So, the first decision variable is the actual ASSIGNMENT, or ORDER, in which the tasks are performed. This is a tactical decision.

The rest of the variables are mainly strategic. The first, medium-range, set of variables in this group is the QUANTITY OF RESOURCES AT EACH LOCATION:

1. Human Resources

2. Assets

Then, in terms of long-range variables, the main variable is the LOCATION of the warehouse and storages.

For this project, the long-term variable LOCATION is set a priori and cannot be changed. The model is prepared ad-hoc for each application to hold a certain number of locations, defined by their geographical coordinates. However, the QUANTITY of resources per location is to be parameterized so that the user might alter the configuration according to the real value before the execution of the application.

The values for the variable ASSIGNMENT are not specified before the execution but they are left for the search engine to find their satisfactory, if not optimal, value.

Output variables

With regard to the output variables, results or criteria, two types might be mentioned. The first type relates to the input- output variable duality. The assigned value to the input variable ASSIGNMENT is an output value of the optimization process, and the decision maker treats it as such, since it is an outcome of the application.

The second type, on the other hand, has to do with the search of the proper value of ASSIGNMENT. For each one of the possible ASSIGNMENT values, a value must be calculated for the measurable criteria, two in this case:

1. TOTAL COST of the ASSIGNMENT, which is to be minimized. The cost is composed of travelling costs as well as of direct labour costs.

2. PERCENT SATISFACTION or SERVICE LEVEL, which is to be maximized. Out of the total tasks, the percentage of maintenance tasks that are completed within the allotted time is calculated.

Additionally, it is necessary to aggregate the two competing criteria in a single measure, which is liable to be optimized. The decision is for the conversion of SERVICE LEVEL into a PENALTY cost, which accounts for the lateness in completing the task. This PENALTY might then be easily included into the TOTAL COSTS as an additional cost.

Data

The first piece of data that is necessary is the NETWORK of LOCATIONS. The LOCATIONS are the physical places where the resources might be located. These places are of three types: locations where the human resource teams rest when idle, locations where the assets are stored when idle, and the locations where the maintenance tasks need to be executed. One two-dimensional table with costs between location and another one with time have to be specified.

For each of the HUMAN RESOURCES, the data that must be a priori specified is: specialization, location while resting, availability and cost per hour. For the ASSETS, the required initial data is: storage location and quantity.

Finally, a table that relates lateness and the penalties must be provided to calculate the PENALTY costs.

THE SIMULATION MODEL

There exist several simulation tools in the market to help in the programming of these scheduling problems. However, it should be reminded that it is necessary to embed the simulation model into the INGRID application, which will act a s the interface. INGRID will include the data, and will present the proposed ASSIGNMENT.

The general-purpose tools like C++ or even Visual Basic seem appropriate to develop the scheduling tool, both at the forecasting level, for different time horizons, and at the constant review level, for on-line reassignment of tasks.

These tools are also appropriate because they present ODBC capabilities with many of the commercial software of any kind and also with INGRID. Even INGRID is able to call and execute routines that have been developed with these general-purpose tools.

With respect to logic and mathematical modelling, the first prototype has been developed in MsExcel Visual Basic. The developed routine includes the following steps:

1. Demand generation or list of tasks to be assigned

2. Reading of input data: network of nodes, resources

3. Random assignment of tasks to teams

4. Awakening of a team which is resting

5. Transportation and execution of the tasks

6. Return to the location to rest

7. Calculation of the values for the different criteria

With respect to statistical modelling of the input data, the variable with a priori probability distributions are:

1. Execution time of the maintenance task

2. Location of the maintenance task

3. Type of maintenance task for specialization purposes.

DECISION MAKING

With the model ready, it is the proper time to develop the decision making routine that adds capabilities for searching for a good assignment of tasks with regard to the TOTAL COST CRITERION. Many different assignments should be generated and evaluated so that the best assignment of tasks is selected.

To perform this experimentation phase, the Genetic Algorithms technique (GA) has been selected, mainly because it has been the most widely used to solve the scheduling problem. Two additional reasons are that they are easily coded and efficiently executed.

Scheduling tool: GA

Genetic algorithms are a systematic way to solve search and optimization problems. The method resembles biological evolution in terms of population, sexual reproduction and mutation, with the philosophy that only the fittest survive.

To apply the GA method, the decision variables are parameterized into a chromosome (x_i,...,x_n); that is, each variable is a gene that might take on several values. For each combination of values, that is for each individual, a cost function f(x_i,...,x_n) is calculated. The final value of each gene will be the number of the task to be performed.

The different codified solutions or individuals compete against the rest to determine which one is the best (but maybe not the optimum since it is usually not feasible to codify and evaluate them all). An elite of the population is maintained, that is, in each generation only those that obtain a better solution survive, reproduce and give the good genes to the offsprings. Genetic diversity is introduced via mutation of genes and crossing of chromosomes.

The method consists in evolving an initial population using crossing patterns between different individuals until a satisfactory evolution of the initial population is obtained. The evolution scheme of a basic GA might be summarized into the following five steps:

1. Randomly initialize a population of solutions to a problem, represented by an adequate variable structure.

2. Evaluate each of the solutions and assign a score which reflects the behaviour of the individual.

3. Choose out of the initial population the part that has a higher score.

4. Cross (combine) and mutate (change) the different solutions of the selected part to rebuild the population to is constant size.

5. Repeat 1 to 4 until the evaluations throughout the population are stable.

Steps 1 and 5 require further discussion: initial population and number of iterations of the algorithms, which are related. The initial population decisively determines the rhythm of convergence of the algorithm. Although convergence to the optimum is a guarantee, the generation of this optimum alternative might take too long.

That is the reason why a lot of attention should be paid to the definition of the initial population. A first set of alternatives might be generated through the assignment of the tasks to the team that is closer to the location where the task is to be performed, favouring service level but not cost. A second set of alternatives, to favours costs, might be generated through the assignment to the cheapest team. Random assignment might be the choice for a third group. A population size of around 20 alternatives is recommended.

Regarding the number of iterations, it might be a priori established as a function of the allotted execution time, once an estimation of the time per iteration is reliably estimated. Additional stopping rules might be implemented. The first relates to the number of consecutive evaluations without improving either the current best alternative or the average population value.

The second deals with the attainment of a satisfactory value. This reference value is usually established in relation with known optimum non-feasible alternatives. In this project, it is possible to calculate the minimum execution cost (that is, all the costs but the penalty). This alternative is one in which each task is performed by the cheapest team whether it is idle or not. Obviously, many tasks will not be finished when needed but an idea of minimum cost is obtained and a reference or satisfactory might be set. If an alternative is evaluated and the value for the total cost is calculated to be below this satisfactory value, the algorithm also stops.

With respect to step 4, more technical, the most common genetic options have been used since they are usually the more efficient. Crossing (Figure 2) is performed mixing two parent chromosomes, which are selected proportionally to the cost value: the lower the value, the higher the probability of being selected to create offsprings. The crossing is performed in two points.

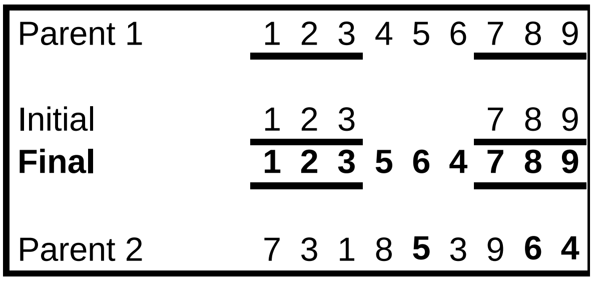

Figure 2: Crossing in GA

Then, a mutation is performed (Figure 3) by randomly selecting one gene and shifting its position, thus shifting the order in which the tasks are executed.

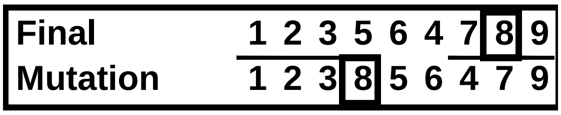

Figure 3: Mutation in GA

THE PROPOSED FRAMEWORK

The previous sections have presented in detailed the different parts of the framework that is going to be used to develop the decision making routines to help in the daily assignments of teams to tasks. Following the modelling approach has allowed for the systematic study of the maintenance system under consideration. The analysis of INGRID has helped in the decision of the software tool to use.

The assignment process might then be correctly represented by the combination of the maintenance software and simulation modelling and combinatorial optimization techniques using the following three steps:

1. Read the input data, like list of remaining tasks and availability and location of the resources, out of INGRID.

2. Read the parameters for the genetic algorithm, including population size and stopping conditions.

3. Execute the simulation-optimization routine that will result in a satisfactory assignment.

CONCLUSION

The framework has been set for the development and implementation of an optimization routine within the commercial software INGRID. The design has called for the utilization of a genetic algorithm due to its very good compromise between efficiency and efficacy.

Therefore, the main general objective explicitly stated within the publicly funded PROFIT program has been met.

REFERENCES

April J., F. Glover, J. P. Kelly; and M. Laguna. 2003. “Practical Introduction to Simulation Optimization.” In *Proceedings of the 2003 Winter Simulation Conference,* New Jersey: IEEE, 71-78.

Boland, L. 1982. *The Foundations of the Economic Method*. George Allen and Unwin, London.

Eglese R.W. 1990. “Simulated Annealing: A Tool for Operational Research.” *European Journal of Operational Research*, vol.46, no.3, 271-281.

Fazlollahi B. and R. Vahidov. 2001. “Extending the Effectiveness of Simulation-Based DSS Through Genetic Algorithms.” *Information and Management*, vol. 39, no.1, 53-65.

Fisher M.L. and C.D. Ittner. 1999. “The Impact of Product Variety on Automobile Assembly Operations: Empirical Evidence and Simulation Analysis.” *Management Science*, vol. 45, no. 6, 771-786.

Goldberg D.E. 1989. *Genetic Algorithms in Search, Optimization and Machine Learning,* New York: Addison-Wesley.

Hershauer J.C. and R.J. Ebert. 1975. “Search and Simulation Selection of a Job-Shop Sequencing Rule.” *Management Science*, vol. 21, no. 7, Theory Series, 833-843.

Hollier R.H. 1968. “A Simulation Study of Sequencing in Batch Production.” *Operations Research*, vol. 19, no. 4, 389-407.

Hopfield J. and D. Tank. 1985. “Neural Computation of Decisions in Optimization Problems.” *Biological Cybernetics*, vol. 52, 141-152.

Iyer S.K. and B. Saxena. 2004. “Improved Genetic Algorithm for the Permutation Flowshop Scheduling Problem.” *Computers and Operations Research*, vol. 31, no.4, 539-606.

Karaboga D. and A. Kalinli. 1997. “A Tabu Search Algorithm for Combinatorial Optimisation Problems.” In *Proceedings of the 1997 European Simulation Multiconference*, SCS.

Macaskill J.L.C. 1973. “Computer Simulation for Mixed-Model Production Lines.” *Management Science*, vol. 20, no. 3, Theory Series, 341-348.

Moccellin J.V. and M.S. Nagano. 1998. “Evaluating the Performance of Tabu Search Procedures for Flow Shop Sequencing.” *The Journal of the Operational Research Society*, vol. 49, no. 12, 1296-1302.

Wein L.M. and P.B. Chevalier. 1992. “A Broader View of the Job-Shop Sequencing Problem.” *Management Science*, vol. 38, no. 7, 1018-1033.

Yu H. and W. Liang. 2001. “Neural Network and Genetic Algorithm-Based Hybrid Approach to Expanded Job-Shop Scheduling.” C*omputers & Industrial Engineering*, vol. 39, no. 3-4, 337-356.

JAVIER OTAMENDI received the B.S. and M.S. degrees in Industrial Engineering at Oklahoma State University, where he developed his interests in Simulation and Total Quality Management. Back in his home country of Spain, he received a B.S. in Business Administration and a Ph.D. in Industrial Engineering. He is currently a simulation and statistics consultant and university professor at the Rey Juan Carlos University in Madrid.