NASA Technical Memorandum 102367

Application of a Hybrid Generation/Utility Assessment Heuristic to a Class of Scheduling Problems

(NASA-TM-102367) APPLICATION OF A HYBRID GENERATION/UTILITY ASSESSMENT HEURISTIC TO A CLASS OF SCHEDULING PROBLEMS (NASA) 13 p CSCL 12B
N90-10674
Unclas
G3/66 0234693

Ann O. Heyward
Lewis Research Center
Cleveland, Ohio

Prepared for the
International Conference on Systems, Man, and Cybernetics
sponsored by the Institute of Electrical and Electronics Engineers
Cambridge, Massachusetts, November 14–17, 1989

NASA

APPLICATION OF A HYBRID GENERATION/UTILITY ASSESSMENT HEURISTIC TO A CLASS OF SCHEDULING PROBLEMS

Ann O. Heyward

NASA Lewis Research Center
Cleveland, Ohio

E-5086

ABSTRACT

A two-stage heuristic solution approach for a class of multiobjective, n-job, 1-machine scheduling problems is described. Minimization of "job-to-job" interference for n jobs is sought. The first stage generates alternative schedule sequences by interchanging pairs of schedule elements. The set of alternative sequences can represent nodes of a decision tree; each node is reached via decision to interchange job elements. The second stage selects the parent node for the next generation of alternative sequences through automated paired comparison of objective performance for all current nodes. An application of the heuristic to communications satellite systems planning is presented.

1. INTRODUCTION

This paper discusses an heuristic approach to solving a class of problems that may be formulated as multiobjective, n-job, 1-machine scheduling problems. The problem has n objectives: the minimization of "job-to-job" interference experienced by n jobs to be scheduled consecutively on one machine. Job-to-job interference may represent a variety of factors which impede scheduling of jobs in immediately adjacent time slots. In a manufacturing environment, such a factor might be the need to remove filings from a work surface between the machining of parts. As noted by Ignizio [4], electromagnetic interference between communications signals may also be modeled as job-to-job interference, as in the application presented here. Improvement of an existing schedule is sought, as opposed to absolute optimality; therefore, an heuristic approach is suitable. A two-stage heuristic procedure is described. The first stage generates alternative schedule sequences via an heuristic swapping procedure, which interchanges pairs of schedule elements (jobs). The set of alternative arrangements that may be derived from an existing arrangement can represent nodes of a decision tree; thus, each node is reachable via a decision to interchange two job elements in the existing arrangement. The first stage of the heuristic enumerates all decision tree nodes reachable from the current arrangement. The second stage of the heuristic selects the most promising potential parent node from those enumerated by the first stage, through automated paired comparison of all current level nodes, for the next generation of alternative schedules.

An extensive body of literature exists for scheduling problems in general (e.g., see Graves' survey [3]) and for n-job, 1-machine scheduling/sequencing problems in particular. Proposed solution techniques have included goal programming [4] and dynamic programming [11] as well as more traditional methods applied to alternative problem formulations [8]. Strategies for reducing the solution space of possible sequences and schedules, or improving efficiency of search algorithms via the exploitation of precedence constraints and/or dominance concepts appear in [1], [2], [5], [9], and [11]. Extensive work has been performed on the use of interactive paired comparison of alternative options to obtain information about preference relationships, as described in [6]. The approach presented in this paper exploits precedence constraints, a particular dominance concept, and automated paired comparison. The technique can thus be said to represent a hybridization of solution generation and utility assessment techniques - approaches that are generally utilized independently of one another.

The heuristic described has been implemented within a module of a computer software package designed for communications satellite systems planning: the Numerical Arc Segmentation Algorithm for a Radio Conference (NASARC) [12,13].

Originally developed at NASA/Lewis Research Center as a planning tool for the 1988 Space World Administrative Radio Conference, the package is designed to develop orbital arc segments that are shared by groups of satellites. In general, satellites allotted to different arc segments will pose a potentially harmful level of interference to one another; thus, the proximity of their respective arc segments will affect the level of interference experienced by satellites within each segment. An exact parallel exists between the problem of developing an arrangement of orbital arc segments that minimizes such segment-to-segment interference, and the n-job, 1-machine scheduling problem where the goal is to minimize the level of job-to-job interference experienced by all jobs.

2. REPRESENTATION OF THE SCHEDULING PROBLEM AS A DECISION TREE

Prior to discussion of the heuristic in more detail, it is appropriate to lay the groundwork for the algorithm by representing the n-job, 1-machine scheduling problem as a decision tree search problem.

The scheduling problem addressed here may be stated in general terms, as follows: given n jobs to be scheduled on one machine, determine the best schedule with respect to one or more optimality criteria, subject to constraints imposed upon the time slots available to each job. If we may treat schedules as sequences, a direct (though not one-to-one) mapping of the set of all possible solutions to a decision tree exists.

Without loss of generality, it is assumed that some initial sequence exists. This initial ordering of jobs may be regarded as the single "parent" node for a decision tree. Each direct descendant of the initial parent sequence will be reached by a single two-job permutation, or interchange, in the order in which jobs are to be performed - similar to that described by Emmons [1], Picard et al. [8], or Reiter et al. [9]. Thus, if our initial schedule is represented by an ordering $(1, 2, \ldots, n)$, the next generation will consist of $\begin{bmatrix} n \\ 2 \end{bmatrix}$ possible schedules. If each node in this generation were to be treated as a potential parent node, the following (second) generation would contain $\begin{bmatrix} n \\ 2 \end{bmatrix}^2$ possible orderings. In general, the k^{th} generation will contain $\begin{bmatrix} n \\ 2 \end{bmatrix}^k$ nodes, each of which represents a possible schedule. It is clear that no loss of generality is experienced in the assumption of an initial ordering; eventually all $(n!)$ possible schedules will each be represented by one or more nodes in the decision tree. This concept is illustrated in Fig. 1.

The three-job example presented in Fig. 1 illustrates some potential drawbacks of this representation of the scheduling problem. Schedules may be duplicated at several nodes, and the decision tree will grow explosively as the number of jobs increases. However, the issue of feasibility has not yet been dealt with. It is expected that some interchanges of jobs within the schedule will be prohibited by virtue of time constraints that restrict when these jobs may be performed, i.e., precedence constraints. In fact, some jobs may be effectively fixed within the schedule if their feasible times are sufficiently restricted. The number of feasible schedules of n jobs for a given problem may thus be significantly less than $(n!)$, as noted by Emmons [1], Erschler et al. [2], and Pasch [7]. Interchanges that are infeasible need not be represented within the decision tree, which will reduce both the total number of nodes and the number of nodes representing duplicate schedules.

The size of the decision tree is also influenced by efficient enumeration of decision tree nodes. Enumerating only those nodes meeting an efficiency criterion will effect further reductions of decision tree size. Efficient enumeration of decision tree nodes is addressed in two ways by the heuristic. First, in any new generation of nodes, only those nodes that represent schedules that improve upon (i.e., dominate) the schedule of the parent node are enumerated (Stage I). Second, enumeration of the next generation of nodes starts from the single most promising (most dominant) parent node of the current generation (Stage II).

The use of dominance concepts in improving efficiency of enumeration is also addressed in Emmons [1], Erschler et al. [2], and Picard et al. [8].

3. HEURISTIC STAGE I: ENUMERATION OF DECISION TREE NODES

We wish to enumerate only those nodes that satisfy criteria for both feasibility and efficiency. Stage I of the heuristic accomplishes this purpose by comparing projected objective function performance of potential new nodes with the objective function values of the parent node. New nodes which dominate the parent node are generated; nodes which do not are discarded. Only nodes which are the product of feasible interchanges are assessed.

Prior to defining feasibility and efficiency conditions, it is appropriate to present a generalized formulation of the scheduling problem we wish to address:

$$\text{Minimize } Z_k = \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j\neq i}}^{n} x_{ij}(x_{j,k-1}+x_{j,k+1})C_{ij} \qquad k=1, \ldots, n \tag{3.1}$$

Subject to:

$$T_{1i} \leq t_{1i}, \quad i=1, \ldots, n \tag{3.2}$$

$$t_{2i} \leq T_{2i}, \quad i=1, \ldots, n \tag{3.3}$$

$$t_{2i} - t_{1i} = r_i, \quad i=1, \ldots, n \tag{3.4}$$

$$\sum_{i=1}^{n} x_{ik}=1, \quad k=1, \ldots, n \tag{3.5}$$

$$x_{jm}x_{ik}(t_{1j}-t_{2j}) \geq 0, \quad i=1,\ldots, n;\ j=1, \ldots, n,\ j\neq i;\ k=1, \ldots, n;\ m=k+1, \ldots, n \tag{3.6}$$

$$x_{ik} = \{0,1\}, \quad i=1, \ldots, n;\ k=1, \ldots, n \tag{3.7}$$

$$t_{2i},\ t_{1i} \geq 0, \quad i=1, \ldots, n \tag{3.8}$$

where

i,j — indices of jobs referenced to positions in initial (parent) sequence

C_{ij} — measure of job-to-job interference experienced by job i as a result of adjacency to job j in the scheduled order of jobs (constant for each pair i,j)

T_{1i} — earliest feasible time at which job i may be begun (constant)

T_{2i} — latest feasible time at which job i may be completed (constant)

t_{1i} start time, job i

t_{2i} completion time, job i

r_i required length of time to perform job i (constant)

x_{ik} {1 if job i is k^{th} in the scheduled order of jobs} {0 otherwise}

The set of n objectives (3.1) consists of minimization of the degree of job-to-job interference experienced by each job. Constraints (3.2) and (3.3) ensure that the time slots found for all jobs i fall within the time limits feasible for each job. Constraint set (3.4) ensures that the time slot scheduled for each job i is of exactly the required length. Constraint set (3.5) ensures that exactly one job i is assigned to each position in the scheduled ordering of jobs. Constraints (3.6) enforce non-overlap of time intervals allotted for jobs 1 to n. Constraints (3.7) enforce integrality of the x_{ik}, and constraints (3.8) enforce nonnegativity.

We define feasibility conditions for the interchange of two jobs within an existing sequence in terms of the problem just presented. An interchange of jobs i and j is feasible if and only if the following three conditions are met:

$$\min(T_{2i},T_{2j}) - \max(T_{1i},T_{1j}) \geq r_i + r_j \tag{3.9}$$

$$\max(t_{2i},t_{2j}) \leq \min(T_{2i},T_{2j}) \tag{3.10}$$

$$\min(t_{1i},t_{1j}) \geq \max(T_{1i},T_{1j}) \tag{3.11}$$

Condition (3.9) guarantees that the intersection of the allowable time intervals for performance of jobs i and j is of at least the length of time required to complete both jobs. Conditions (3.10) and (3.11) ensure that the intersection of the feasible time intervals for jobs i and j encompasses both of the time intervals allotted to jobs i and j. Note that a slightly different interpretation may be given to the variables t_{1i}, t_{2i}, t_{1j}, and t_{2j}: the values of these variables define the <u>current</u> time intervals allotted to jobs i and j. Thus, compliance with conditions (3.10) and (3.11) implies that the current time slots for jobs i and j may be interchanged in order and still remain within feasible limits in both cases. Conditions (3.9) to (3.11) are illustrated in Fig. 2.

Conditions (3.9) to (3.11) may be applied to all pairs of jobs i and j to construct a matrix of feasible interchanges. The matrix will be square, but nonsymmetric. Each column corresponds to a job i, and contains the jobs j which may be interchanged with job i. Matrix entries are defined as follows:

$a_{j,i}$ {j if job j is interchangeable with job i} {0 otherwise}

By examining the matrix, we can restrict enumeration of nodes to those that are the product of feasible interchanges.

In examining the problem defined by (3.1) to (3.8), we observe that any feasible solution will have a corresponding objective vector containing n elements. Our definition of efficiency will be based upon element-by-element comparison of objective vectors, but first it is appropriate to more clearly define the objectives to be calculated.

Measures of various types of interference between two elements may be expressed in terms of the separation between the two elements required to reduce the level of interference to an acceptable level or to eliminate it completely. A measure of this type is often preferable to a rigorous computation of interference, as expressions for interference are often nonlinear. If done on a worst-case basis, measures of this type have the additional advantage of requiring a one-time a priori calculation, rather than continual reevaluation of a complicated expression.

We will assume that we have available (or can readily calculate) a matrix of required minimal time separations between all pairs of jobs i and j, calculated on the assumption that jobs i and j are adjacent in the scheduled order of jobs. The required time separation will thus be worst-case. The elements of this matrix will provide the a_{ij} of (3.1), i.e., a_{ij} will be the time required between finishing job i and starting job j if jobs i and j occupy adjacent positions in the scheduled order.

The efficiency of an interchange is evaluated by calculating the resulting change in affected objective function values. For a given interchange of two jobs i and j, in sequence positions k and m, up to six objective function values may be affected, namely, those objective values for jobs assigned to positions $k-1$, $k,k+1$, $m-1$, m, and $m+1$ (see Fig. 3). Assuming that the goal is to improve the objective value for position k, the new objective value obtained when job i is interchanged with job j is calculated first. If the objective value is degraded, the interchange is rejected. If the objective value is improved, the magnitude of the improvement is calculated to be used as a standard of reference. Remaining affected objective values are calculated. If any value is degraded as a result of the interchanges of jobs i and j, the degradation must be of lower magnitude than the degree of improvement gained in objective i. If this criterion is not met, the interchange is rejected.

If the interchange is efficient, generation of a new node (sequence) is attempted. If the length of time required for jobs in sequence positions k and m are equal, a direct swap of their slots in the sequence may take place. It is possible, if the jobs require different lengths of time, that one available time slot will be of insufficient length to allow direct insertion of a longer job. In this case, a simple algorithm may be applied to laterally shift adjacent jobs to slightly later or earlier time slots to open a time slot of sufficient length. This process is illustrated in Fig. 4.

If a time slot of sufficient length cannot be created for the longer job, the interchange is rejected. If the longer job can be scheduled, the new sequence is generated and stored for comparison with other nodes of the current generation. The heuristic will then evaluate the next feasible interchange for job i, applying the same efficiency tests.

Assuming a priori generation of feasible interchange and interference measure matrices, Stage I of the heuristic may be described as follows:

- Step 1: Determine next candidate job to be utilized for node generation (job i). If candidates are exhausted, stop.
- Step 2: By examining column i of the feasible interchange matrix, select next candidate job j for possible interchange with job i. If candidates are exhausted, go to Step 1.
- Step 3: Calculate objective value for k^{th} slot resulting from substitution of job j for job i. If objective value is degraded, update interchange matrix and go to Step 2. Otherwise, calculate improvement gained.
- Step 4: Calculate objective value for slots $k-1$, $k+1$, $m-1$, m, and $m+1$. If, for any objective value, degradation is not offset by improvement in objective value for k^{th} slot (Step 3), update interchange matrix and go to Step 2.
- Step 5: If required times for performance of jobs i and j are equal, interchange time slots and generate new ordering. Update interchange matrix and go to Step 2. If times are unequal, go to Step 6.
- Step 6: Calculate time available between jobs occupying slots adjacent to shorter job. If time available is of sufficient length for longer job, allocate time slots and generate new ordering. Update interchange matrix and go to Step 2. If time available is insufficient, go to Step 7.
- Step 7: Attempt lateral shift of jobs adjacent to desired time slot to earlier/later starting time to open slot of sufficient length. If lateral shift fails, update interchange matrix and go to Step 2. If lateral shift succeeds, allocate time slots and generate new ordering. Update interchange matrix and go to Step 2.

A further efficiency enhancement may be realized through generation of an efficient initial parent node for the decision tree. This is accomplished via application of Stage I. The matrix of feasible interchanges may indicate that selected pairs of schedule elements are interchangable only with each other; such elements are termed "pair-restricted". An important feature of such interchanges is that their execution can affect no other outcome of interchanges. Thus, if a pair-restricted interchange improves the initial schedule supplied to the algorithm, a more efficient starting sequence is generated. Generation of an efficient starting node may take place prior to a more general examination of the interchange matrix for all i, as described above. However, straightforward, successive applications of Stage I to the initial schedule will also result in the eventual acceptance of all efficient pair-restricted interchanges.

4. HEURISTIC STAGE II: SELECTION OF PARENT FOR NEXT GENERATION OF NODES

While the steps taken in Stage I of the heuristic reduce the number of nodes in the scheduling decision tree by limiting nodes to those representing only feasible and efficient sequences, it is still possible that a given generation of nodes will contain a large number of members, particularly if the number of jobs to be scheduled is large or if there are few precedence constraints. Thus, to further reduce the size of the scheduling decision tree and yet still obtain a good-quality schedule, Stage II of the heuristic selects the dominant node of the current generation. Stage I of the heuristic is then applied to this node in creating the next generation of possible sequences.

Selection of the dominant parent node is performed through automated paired comparison of schedules. Extensive work has been carried out by Malakooti [6] on the use of interactive paired comparison between alternatives in the assessment of utility functions. However, automated paired comparison is appropriate for this problem because the quality of one schedule versus that of another may be readily assessed by a single quantitative utility measure. Preference results will be transitive, guaranteeing that selection of the dominant parent node through paired comparison will not eliminate a node that might have been preferable to our final choice.

Stage II of the heuristic is performed after each completed application of Stage I; i.e., after a new feasible, efficient node has been generated. The new node will be compared with the best node found so far, on the basis of a numeric index of solution utility. If the newest node is dominant, the existing best node will be replaced. At the beginning of each new generation of nodes, the current best node will be the dominant parent found in the prior generation. At the end of each new generation of nodes, the current best node will be the dominant parent for the next generation.

Solution utility is assessed on the basis of comparative objective function performance. The objective vector of the existing best node and the objective vector of its potential replacement are compared on an element-by-element basis. Since our goal is to reduce job-to-job interference in all n time slots, a simple measure is defined for overall solution utility:

$$Q_{jk} = \sum_{i=1}^{n} \delta_{ijk} \qquad j,k=1,2,j\neq k$$

where

δ_{ijk} {1, if the i^{th} component of the objective is lower in value for solution j than the corresponding objective component for solution k}
{0 otherwise}

Q_{12} thus provides a count of the number of objective values that are lower for solution 1 (the best solution so far) than their counterparts for solution 2 (the new node). Q_{21} provides a similar index for the new node versus the best node found so far. The solution that is preferred is that for which the maximum of Q_{12} and Q_{21} is attained, i.e., the solution that corresponds to a greater number of lower objective values. It is possible that a tie will result. In that case, the solution with the lowest average objective value over n components is considered dominant. In the unlikely event that a tie results for this preference measure, the existing best solution is retained.

The complete heuristic, incorporating both Stages I and II, may now be stated:

- Step 1: Generate a matrix of interference measures, for all possible adjacent pairings of jobs.
- Step 2: Generate a matrix of feasible interchanges from an existing schedule, using conditions (3.9) to (3.11).
- Step 3: Examine matrix for pair-restricted interchanges. If none are found, continue. If pair-restricted interchanges exist, apply Stage I to create efficient starting sequence. If initial sequence cannot be improved, starting sequence is initial node supplied to heuristic.
- Step 4: Update interchange matrix, eliminating both accepted and rejected interchanges.
- Step 5: If the matrix is nonempty, continue. If no interchanges are feasible, stop. Best feasible node has been found.
- Step 6: Apply Stage I to create a new node (sequence).
- Step 7: Compute solution utility indices Q_{12} and Q_{21} for the best solution to date, and its potential replacement. If $Q_{12} \neq Q_{21}$, go to Step 9.
- Step 8: Compute the average objective function value, over the n components of the objective vector, for the sequences represented by each node.
- Step 9: On the basis of the preference criterion, select the dominant solution that will become the best solution found to date, of the current generation.
- Step 10: If the interchange matrix has not yet been examined for all jobs i, go to Step 5. Otherwise, the current generation is complete; go to Step 4.

5. APPLICATION OF THE HEURISTIC WITHIN THE NASARC SOFTWARE PACKAGE

The Numerical Arc Segmentation Algorithm for a Radio Conference (NASARC) is a software package developed at the NASA/Lewis Research Center as a tool for use at the 1988 World Administrative Radio Conference for allotment planning of satellites in the expansion frequency bands of the Fixed Satellite Service. NASARC utilizes heuristic algorithms to produce a collection of orbital arc segments, each shared by a group of "compatible" satellite systems. Systems are compatible if they exert relatively little interference upon one another in spite of minimal orbital separation. Each group of compatible satellites, and its associated arc segment, may be regarded as a single unit. Each such unit must be placed - or "scheduled" - in some portion of the 360-degree geostationary orbital arc.

The geostationary orbital arc may be regarded as a finite continuous interval within which a number of arc segments of varying length must be accommodated. Satellite systems exert a degree of electromagnetic interference upon one another that is largely dependent on the proximity of their respective positions in the geostationary orbit in relation to the geographical proximity of their service areas. Thus, systems associated with a given segment of the orbital arc interfere to varying degrees with systems associated with all other segments. Segment-to-segment interference can thus be quantified in terms of the interference between systems associated with each segment. The arc segment placement problem is therefore directly analogous to the n-job, 1-machine scheduling problem, where the objective is to minimize the job-to-job interference experienced by each of the n jobs.

In order to improve the arrangement of such segments within the geostationary arc, the heuristic described in previous sections was implemented in a module of the NASARC software package. The module seeks an arrangement of arc segments that results in minimal interference between systems in adjacent segments.

The degree of interference experienced by systems within a given arc segment (with respect to systems in immediately adjacent segments) can be evaluated via a required separation measure of the type described in Section 3. Segment-to-segment interference is characterized as the normalized orbital separation required between systems in adjacent orbital arc segments as follows:

$$E_I\ (W_I) = \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} s_{ij}}{M^*N}$$

where:

E_I or W_I	interference measure associated with arc segment immediately adjacent to the East (following) or West (preceding) of segment I
M	the number of systems in segment I
N	the number of systems in segment I+1 or I-1
s_{ij}	orbital separation required between satellites i and j

The quantities W_I and E_I are equivalent to the quantities a_{ij} as defined in Section 3, and are determined by calculation prior to application of the heuristic. The heuristic will seek improvement in n objectives for n arc segments, i.e.,

$$Z_I = E_I + W_I,\ I=1,\ \dots,\ n$$

which are similar to the set of objectives defined by (3.1).

An initial arrangement of arc segments is determined prior to application of the heuristic. The heuristic then interchanges locations of pairs of arc segments (and their associated satellite systems), in an attempt to decrease interference between systems in adjacent portions of the orbital arc.

A matrix of feasible segment interchanges is first created. An interchange of segments is feasible if and only if the two segments meet the following conditions:

(1) The feasible arc locations associated with each segment intersect by at least the sum of the required arc lengths for each segment.

(2) The intersection of the feasible arc locations associated with each segment encompass both segments' currently allotted arc locations.

These conditions are those defined formally by (3.9) to (3.11), but have an additional physical interpretation for this application. Condition (1) restricts our consideration of alternatives to meaningful alternatives. If condition (1) is not met, the two segments are restricted to portions of the orbit that are sufficiently distant to imply that the appropriate service areas cannot be served by appropriate satellites if the locations of the segments are interchanged. Condition (2) ensures that we may directly interchange segments in our arrangement, subject to minor adjustments for slightly different length requirements for the two segments.

The process described in Sections 3 and 4 is then applied to the arc segment arrangement problem. The matrix of feasible interchanges is examined for pair-restricted interchanges. Those resulting in an improved arrangement are carried out, and an efficient starting solution is generated. Examination of the updated interchange matrix then begins and new arrangements are generated (Stage I) and evaluated via paired comparison (Stage II). Finally, when no further interchanges are possible, the improved arrangement is output as the final arrangement of arc segments derived by the NASARC software package.

A short computational example illustrates the use of the algorithm within the NASARC software. The NASARC package consists of four program modules with the above algorithm residing within the fourth and final module. For the purpose of comparing computation times and results associated with the heuristic, an alternate fourth module was constructed with all features of the heuristic disabled. The complete package, both with and without the heuristic, was then applied to a scenario typical of those for which the NASARC software is utilized.

A comparison of results obtained with and without the heuristic demonstrates that improvement in objective values is obtained at comparatively little computational expense. An additional 25.43 CPU seconds on an Amdahl 5860 running under the VM operating system was required when the heuristic was utilized. This time represents a somewhat conservative measure of the speed of the heuristic, since both computational and noncomputational (i.e., reporting) features associated with the heuristic were disabled within the alternate module. Resulting arrangements of arc segments within the geostationary arc, and the objective values associated with each, are presented in Table 1.

Recalling that the objective is one of minimizing an interference measure associated with each arc segment, Table 1 illustrates that objectives for segments 3, 4, 15, 16, 17, 23, and 24 are improved by application of the heuristic; objectives for segments 5, 6, 18, 19, 20, and 25 are slightly worsened. However, the average improvement in objective value is 0.76, which more than offsets the average degradation of 0.23. Examination of results obtained with the heuristic also demonstrates that a degree of "levelling" of objective values takes place. This is due to the fact that the heuristic allows an interchange of two schedule elements to proceed as long as the improvement gained in the objective being examined offsets the possible degradation of any other single objective value. This effect is desirable for this application, in which scarce orbital arc resources must be distributed as equitably as possible.

6. CONCLUSIONS

An heuristic approach to solution of a class of multiobjective n-job, 1-machine scheduling problems has been presented. The scheduling problem is formulated as a decision tree search problem, with alternative schedules represented as alternative orderings of jobs. Alternative orderings are represented by nodes of a decision tree. Each node is reachable through feasible and efficient interchange of two schedule elements. The heuristic approach to generation and utility assessment of decision tree nodes is two-stage. Stage I of the heuristic ensures that only feasible and efficient (improved) nodes are generated. Stage II of the heuristic ensures that only the most promising parent node of the current generation of nodes is selected for application of Stage I in creation of the next generation of nodes.

The heuristic has been implemented in a module of the Numerical Arc Segmentation Algorithm for a Radio Conference (NASARC), a software package developed for satellite systems planning purposes. The problem formulation and heuristic have important advantages in this application. The problem formulation allows substantial simplification of interference relationships and solution approach. The complexity of calculating interference relationships for a large number of satellite systems is avoided through the use of worst-case required separations as an interference measure in calculating objective function values. While this approach is certainly not unique to this formulation (i.e., see [5]), it is an important feature. It is expected that this feature of the problem formulation may be applicable to a wide variety of scheduling problems where minimization of job-to-job interference is sought. The heuristic is well suited to this type of problem because we seek improvement of an initial solution rather than absolute optimality. The heuristic allows a limited degree of automated, rule-based decision-making appropriate to a multiobjective problem via formulation of improvement measures for Stage I and formulation of preference criteria in Stage II. Since the heuristic treats the scheduling problem as an ordering problem, variables describing exact positions of each schedule element need not be dealt with, other than to the extent needed in making minor adjustments to accomplish an interchange. The heuristic also tends to level objective function values, which is a desirable feature in cases where equitable treatment of schedule elements (in terms of objective function value achieved) is a concern. The heuristic provides an easily implemented and efficient means of achieving the goal of an improved solution, without dependence on system-resident optimization packages. This feature was particularly important for NASARC, which was installed on a variety of computer systems throughout the world and was required to produce consistent results over all systems on which it was implemented. It is expected that these advantages would apply in the solution of scheduling problems for a variety of other applications.

REFERENCES

[1] H. Emmons, "One-Machine Sequencing to Minimize Certain Functions of Job Tardiness," Oper. Res., vol. 17, no. 4, pp. 701-715, July-Aug. 1969.

[2] J. Erschler, G. Fontan, C. Merce, and F. Roubellat, "A New Dominance Concept in Scheduling n Jobs on a Single Machine with Ready Times and Due Dates," Oper. Res., vol 31, no. 1, pp. 114-127, Jan.-Feb. 1983.

[3] S.C. Graves, "A Review of Production Scheduling," Oper. Res., vol. 29, no. 4, pp. 646-675, July-Aug. 1981.

[4] J.P. Ignizio, "A Generalized Goal Programming Approach to the Minimal Interference, Multicriteria Nx1 Scheduling Problem," IEEE Trans., vol. 16, no. 4, pp. 316-322, Dec. 1984.

[5] E.L. Lawler, "Sequencing Jobs to Minimize Total Weighted Completion Time Subject to Precedence Constraints," in Annals of Discrete Mathematics 2. New York: North-Holland Publishing Co., 1978, pp. 75-90.

[6] B. Malakooti, "Theories and an Exact Interactive Paired-Comparison Approach for Discrete Multiple-Criteria Problems," IEEE Trans. on Syst. Man Cybernetics, vol. 19, no. 2, pp. 365-378, Mar.-Apr. 1989.

[7] K.A. Pasch, "Heuristics for Job Shop Scheduling" Ph.D. Thesis, Massachusetts Inst. of Technology, 1988.

[8] J. Picard and M. Queyranne, "The Time-Dependent Traveling Salesman Problem and Its Application to the Tardiness Problem in One-Machine Scheduling," Oper. Res., vol. 26, no. 1, pp. 86-110, Jan.-Feb. 1978.

[9] S. Reiter and G. Sherman, "Discrete Optimizing," SIAM J., vol. 13, no. 3, pp. 864-889, Sept. 1965.

[10] C.H. Reilly, "A Satellite System Synthesis Model for Orbital Arc Allotment Optimization", IEEE Trans. Commun., vol. 36, no. 7, pp. 845-849, July 1988.

[11] L. Schrage and K.R. Baker, "Dynamic Programming Solution of Sequencing Problems with Precedence Constraints," Oper. Res., vol. 26, no. 3, pp. 444-449, May-June, 1978.

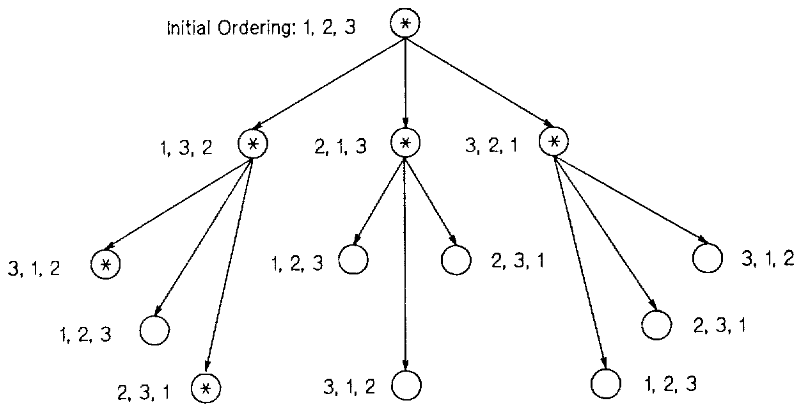
[12] W.A. Whyte, A.O. Heyward, D.S. Ponchak, R.L. Spence, and J E. Zuzek, "Numerical Arc Segmentation Algorithm for a Radio/Conference - NASARC (Version 2.0) Technical Manual," NASA TM-100160, 1987.

[13] W.A. Whyte, A.O. Heyward, D.S. Ponchak, R.L. Spence, and J.E. Zuzek, "Numerical Arc Segmentation Algorithm for a Radio Conference - NASARC (Version 2.0) User's Manual," NASA TM-100161, 1987.

TABLE 1

Arc Segment Arrangements

Heuristic Not Utilized				Heuristic Utilized				Change in Objective
Segment Index	Segment Location (Longitude)		Objective Value	Segment Index	Segment Location (Longitude)		Objective Value	
1	-168.00	-164.00	1.05	1	-168.00	-164.00	1.05	-
2	-119.00	-115.00	1.11	2	-119.00	-115.00	1.11	-
3	-115.00	-111.00	2.35	3	-115.00	-111.00	2.19	-0.16
4	-111.00	-107.00	2.26	5	-111.00	-106.00	2.11	0.28
5	-107.00	-102.00	1.83	4	-106.00	-102.00	1.92	-0.34
6	-102.00	-96.00	1.47	6	-102.00	-96.00	1.56	0.09
7	-96.00	-92.00	1.69	7	-96.00	-92.00	1.69	-
8	-88.00	-83.00	1.69	8	-88.00	-83.00	1.69	-
9	-78.00	-74.00	2.03	9	-78.00	-74.00	2.03	-
10	-74.00	-70.00	2.30	10	-74.00	-70.00	2.30	-
11	-70.00	-66.00	1.74	11	-70.00	-66.00	1.74	-
12	-66.00	-62.00	0.98	12	-66.00	-62.00	0.98	-
13	-62.00	-59.00	0.19	13	-62.00	-59.00	0.19	-
14	-58.00	-55.00	0.22	14	-58.00	-55.00	0.22	-
15	-50.00	-44.00	2.10	15	-50.00	-44.00	0.76	-1.34
16	-44.00	-41.00	3.18	17	-44.00	-40.00	1.34	-0.76
17	-41.00	-35.00	2.10	18	-40.00	-37.00	1.94	0.30
18	-35.00	-29.00	1.64	16	-37.00	-31.00	1.96	-1.22
19	-29.00	-25.00	1.40	19	-31.00	-25.00	1.48	0.08
20	-25.00	-16.00	1.20	20	-25.00	-16.00	1.29	0.09
21	-16.00	-9.00	1.18	21	-16.00	-9.00	1.18	-
22	-9.00	0.00	1.58	22	-9.00	0.00	1.58	-
23	0.00	3.00	2.92	23	0.00	3.00	2.44	-0.48
24	3.00	8.00	2.95	25	3.00	10.00	2.47	0.54
25	8.00	15.00	1.93	24	10.00	15.00	1.94	-1.01
26	15.00	19.00	1.46	26	15.00	19.00	1.46	-
27	19.00	26.00	1.33	27	19.00	26.00	1.33	-
28	26.00	30.00	1.67	28	26.00	30.00	1.67	-
29	30.00	35.00	1.90	29	30.00	35.00	1.90	-
30	35.00	42.00	1.90	30	35.00	42.00	1.90	-
31	42.00	46.00	1.75	31	42.00	46.00	1.75	-
32	46.00	54.00	1.86	32	46.00	54.00	1.86	-
33	54.00	57.00	2.46	33	54.00	57.00	2.46	-
34	60.00	65.00	2.71	34	60.00	65.00	2.71	-
35	65.00	71.00	2.10	35	65.00	71.00	2.10	-
36	71.00	79.00	1.52	36	71.00	79.00	1.52	-
37	80.00	86.00	1.25	37	80.00	86.00	1.25	-
38	86.00	95.00	1.49	38	86.00	95.00	1.49	-
39	95.00	98.00	2.95	39	95.00	98.00	2.95	-
40	98.00	102.00	3.60	40	98.00	102.00	3.60	-
41	106.00	112.00	1.74	41	106.00	112.00	1.74	-
42	112.00	115.00	0.30	42	112.00	115.00	0.30	-
43	115.00	119.00	1.27	43	115.00	119.00	1.27	-
44	120.00	125.00	3.01	44	120.00	125.00	3.01	-
45	125.00	129.00	3.47	45	125.00	129.00	3.47	-
46	129.00	138.00	2.57	46	129.00	138.00	2.57	-
47	138.00	141.00	2.03	47	138.00	141.00	2.03	-
48	151.00	158.00	1.99	48	151.00	158.00	1.99	-
49	158.00	164.00	2.02	49	158.00	164.00	2.02	-



(starred nodes indicate first occurrence of one of 3! possible orderings)

FIGURE 1. - DECISION TREE REPRESENTING ALL POSSIBLE SEQUENCES OF THREE JOBS.

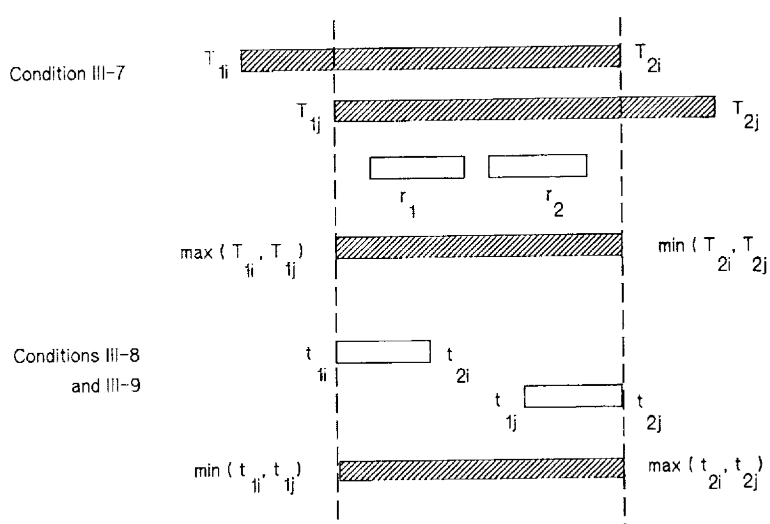


FIGURE 2. - FEASIBILITY CONDITIONS FOR ALLOWABLE INTERCHANGE OF JOBS WITHIN SCHEDULE.

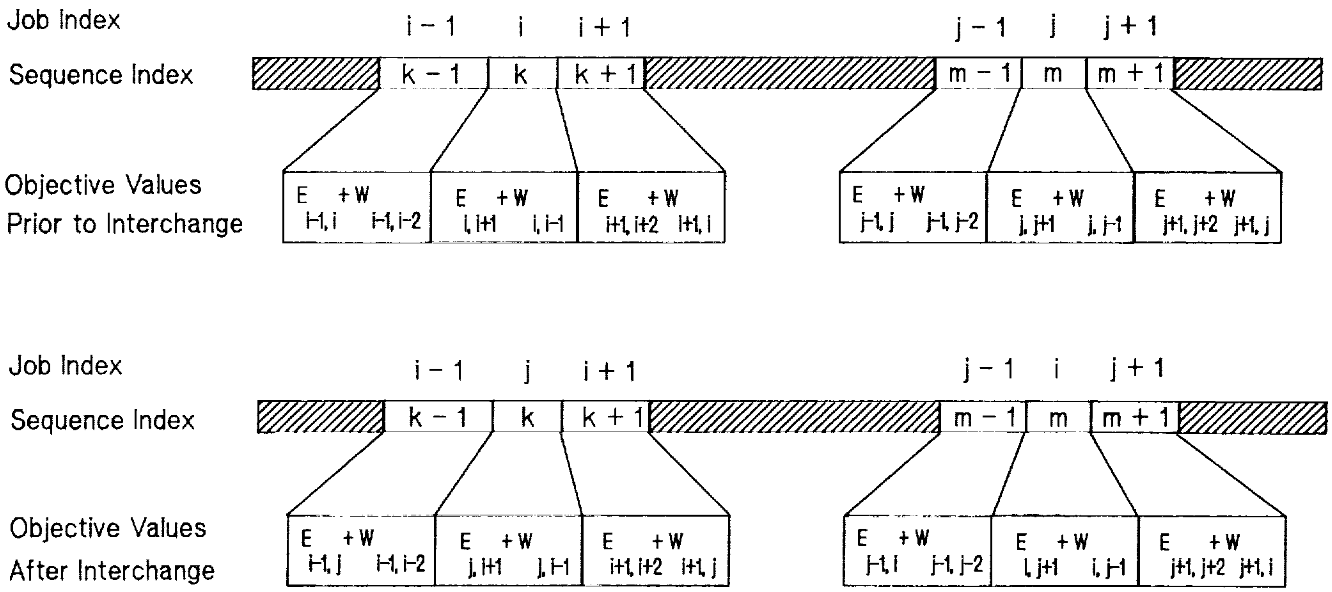


FIGURE 3. - OBJECTIVE VALUES AFFECTED BY INTERCHANGE OF JOBS WITHIN SCHEDULE.

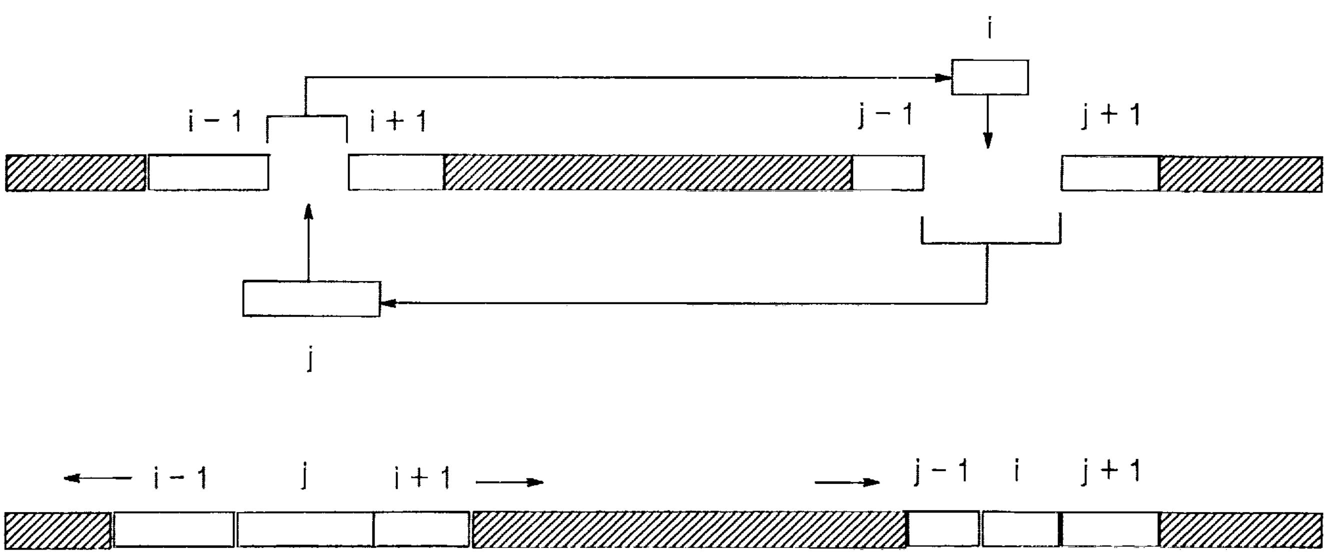


FIGURE 4. - INTERCHANGE OF SCHEDULE ELEMENTS ACCOMPANIED BY LATERAL SHAFT OF ADJACENT ELEMENTS.

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No. NASA TM-102367	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Application of a Hybrid Generation/Utility Assessment Heuristic to a Class of Scheduling Problems		5. Report Date September 1989
		6. Performing Organization Code
7. Author(s) Ann O. Heyward		8. Performing Organization Report No. E-5086
		10. Work Unit No. 643-10-01
9. Performing Organization Name and Address National Aeronautics and Space Administration Lewis Research Center Cleveland, Ohio 44135-3191		11. Contract or Grant No.
		13. Type of Report and Period Covered Technical Memorandum
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546-0001		14. Sponsoring Agency Code

15. Supplementary Notes

Prepared for the International Conference on Systems, Man, and Cybernetics, sponsored by the Institute of Electrical and Electronics Engineers, Cambridge, Massachusetts, November 14-17, 1989

16. Abstract

A two-stage heuristic solution approach for a class of multiobjective, n-job, 1-machine scheduling problems is described. Minimization of "job-to-job" interference for n jobs is sought. The first stage generates alternative schedule sequences by interchanging pairs of schedule elements. The set of alternative sequences can represent nodes of a decision tree; each node is reached via decision to interchange job elements. The second stage selects the parent node for the next generation of alternative sequences through automated paired comparison of objective performance for all current nodes. An application of the heuristic to communications satellite systems planning is presented.

17. Key Words (Suggested by Author(s)) Scheduling Heuristic Multiple objectives	18. Distribution Statement Unclassified – Unlimited Subject Category 66

19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No of pages 12	22. Price* A03

NASA FORM 1626 OCT 86

*For sale by the National Technical Information Service, Springfield, Virginia 22161