D39

N79-15627

A MODEL FOR DYNAMIC ALLOCATION OF HUMAN ATTENTION AMONG MULTIPLE TASKS +

Thomas B. Sheridan and M. Kamil Tulga
Man-Machine Systems Laboratory
Department of Mechanical Engineering
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

Abstract

This paper consists of two parts. The first part describes the problem of multi-task attention allocation with special reference to aircraft piloting, the experimental paradigm we use to characterize this situation and the experimental results obtained in the first phase of our research. A qualitative description of an approach to mathematical modeling, and some results obtained with it are also presented to indicate what aspects of the model are most promising. The second part of the paper consists of two appendices which (1) discuss the model in relation to graph theory and optimization and (2) specify the optimization algorithm of the model.

1. Introduction

We think that an increasingly crucial aspect of piloting an aircraft is "multi-task allocation of attention". The pilot must monitor many more systems than before, most of which are growing in complexity. In earlier days flying the aircraft "by the seat of the pants" was difficult, but piloting was, more or less, a constant task. It was obvious that the pilot could keep track of what was being controlled at what time and how well that was working because he was doing it; he was in the loop and could see or feel it directly.

As systems become automatic the pilot himself tends to lose track of what signals are coming into what subsystem and what response that subsystem is making. Most of the time when everything is normal the automatic systems do just fine. Indeed if we demanded that the pilot actually perform all functions which are now automated it is clear he couldn't do a fraction of such tasks. Yet we expect him to monitor all such functions, and at the first overt alram or even subtle evidence of failure we expect him to be able to render a quick accurate diagnosis of the problem and set it straight.

We call the pilot a "flight manager" or "supervisory controller" and we see him in the image of a corporation manager with legions of dutiful automatic servants doing his will and bringing him information as he desires it. The problem is that the corporate manager has time to ponder and investigate and weigh evidence and consider his decisions. He operates

+ research supported by NASA Grant NSG 2118.

568

PAGE INTENTIONALLY BLANK

on a human time scale: if the corporation manager sees his "production vehicle" about to go bankrupt he has at least a few minutes to decide what's wrong and what to do about it. The flight manager doesn't.

The general research questions implied are:

a) What are the expected behaviors and what are the limits of a person's capability to allocate his attention among many simultaneous tasks of varying importance and varying urgency, as a function of the number of tasks, the general pace at which they occur and other salient parameters?

b) If there is a normative or optimal way a person should perform such a task, can it be specified as a quantitative model, and how close does a trained person come to behaving optimally?

c) What are the implications for improving the design of the man-machine systems in which the pilot must perform such multi-task allocation decisions?

2. Experimental Paradigm

To characterize such a multi-task decision-making situation we have developed a very general experimental paradigm and an associated model. The experimental paradigm requires the subject (or decision-maker DM) to select one at a time from among a number of blocks ("tasks") of different heights and widths displayed simultaneously on a CRT (Figure 1). His selection, made by holding a cursor even with the block "attended to" is in order to maximize his reward, where the earning rate is proportional to the displayed "importance" (indicated by the height of each block) and the "productivity rate" (the rate at which the block decreases in width when "attended to"). Blocks appear at random distances from a "deadline" and move at constant velocity toward that deadline, disappearing when they first touch it. Various task parameters have to do with the frequency at which new blocks appear, the speed with which they move toward the deadline, the variability in importance, the variability in how far from the deadline they first appear, and so on. The toal is to "remove" as much block area as possible.

In one experiment blocks continually appear with exponential distribution in time. In a second experiment all blocks appear at the start of the run; no new ones appear thereafter.

An important feature of the experiment is that blocks do not queue up for service, i.e., if a block reaches the deadline the opportunity to earn its reward is lost. We cannot say for sure, however, whether blocks queue in the operator's mind for attention in correspondence to the fact that at any one instant of time there may be some blocks which are far from the deadline and others which are close. The close ones, of course, may be of little importance, so often it is better to attend to more important tasks which are farther from the deadline in order to ensure that all of the really important ones do get attended to before the deadline.

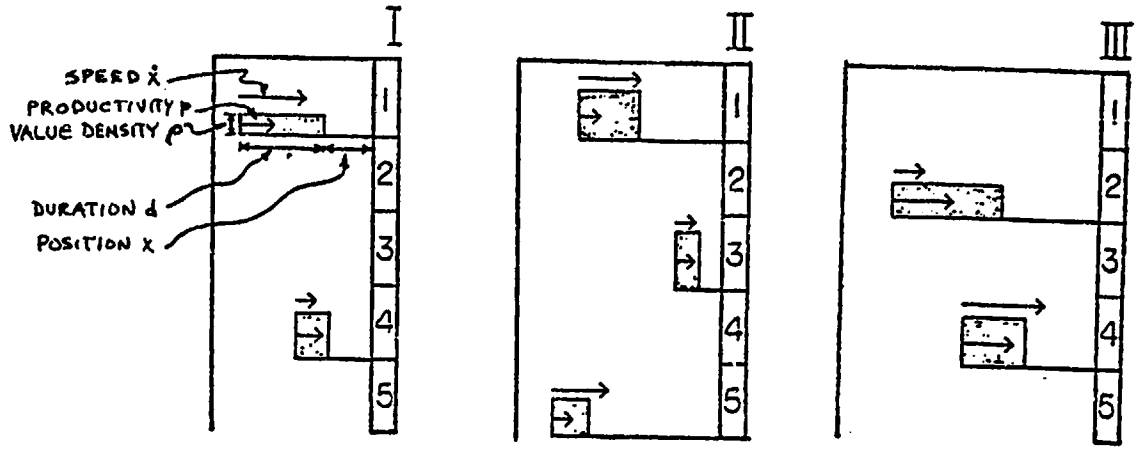

Figure 1. Experimental Computer Display of Moving Blocks Representing Tasks

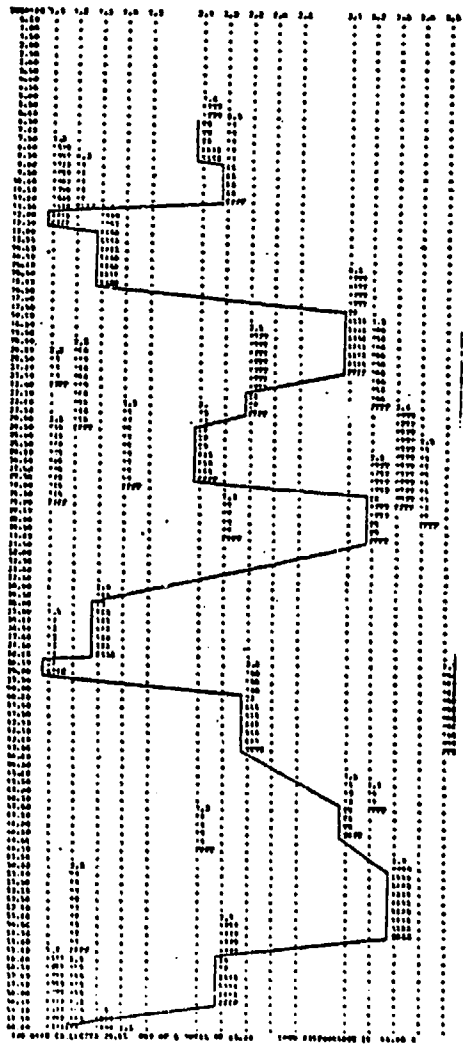

Figure 2. Record of D.M.'s Action Trajectory in Real Time

Figure 2 illustrates a means we have used to obtain a time-plot of which block the subject selects in which queue (column headings). Printed symbols in each column tell the service time the block requires, the time the block will be available, and the value to be obtained.

Having informally experimented with this situation with a variety of parameter combinations we are now in a position to claim that the experiment does seem to simulate various attentional demands which are placed on the pilot. These vary considerably in duration. Some tasks are urgent, but of modest importance; some are urgent and of great importance; some are not urgent and of modest importance; some are not urgent but of great importance to be done before the deadline.

3. Experimental Results

As the first phase of the second author's doctoral thesis, experiments with human subjects have been run with various experimental parameter combinations. Because the number of such possible combinations is so large we have investigated the effects of changing one parameter at a time, relative to a "baseline condition". Table 1 indicates that for all runs the subject worked with 3 queues of blocks (tasks) and runs lasted 400 seconds. The baseline parameters are given above. Seven changes in parameters are indicated below, made one run at a time, all other parameters matching the baseline condition in each case. For each the values gained by each of three subjects, the range of their data, the average, and the total possible are given.

In Table 1 it is seen that a considerably higher speed of blocks moving toward the deadline (2) reduces the score, but not much, compared to the baseline (1). Greater variation in block speed (3) makes little difference. A reduction of interarrival time (4) of blocks means more blocks become available - more opporunity is there for earning a score - but a smaller fraction of these are completed. As the height of blocks (task value densities) become more variable (5) the net earnings are little affected, though the presence of a few very lucrative blocks doubles the total possible score. Giving partial credit (6) for productivity (allocation) on a task when it hits the deadline increases the earnings little more than one percent, which is surprising. Lowering productivity (7) has the most significant effect, as seems intuitively reasonable - but the reduction in score is not quite in proportion to the forced reduction in rate of doing tasks.

4. A Mathematical Modeling Approach

To accompany the experimental task, we have developed a mathematical model which can be run on the computer immediately after any human data run. (The relationship of the model to graph theory in general and the full specification of the model algorithm given in Appendices 1 and 2 respectively).

COMMON CONDITIONS

3 queues, 400 sec duration

BASELINE CONDITION

Task Interrival time, exponential distribution, mean = 20 sec/queue

all tasks appear 5 units away from the deadline

all tasks 2.5 units in duration

all tasks speed toward deadline at 0.1 units/sec.

productivity on all tasks 0.5 units per sec.

value density rectangular distributor 0 - 1 utiles/sec

No partial credit was given in the baseline case.

	CONDITION	% AVAILABLE VALUE GAINED BY SUBJECTS DY	KT	SJ	RANGE	AVG.	TOTAL POSSIBLE VALUE (UTILES)
1	Baseline, B	.913	.931	.942	.029	.929	98.7
2	More speed (2.5 B)	.917	.880	.878	.061	.891	98.7
3	Variable speed (rect, .05-2.5)	.934	.907	.912	.027	.918	98.7
4	Less interarrival time (0.75B)	.803	.809	.795	.014	.802	122.2
5	More varied value density (rect dist 0-2)	.946	.940	.902	.044	.929	197.6
6	Baseline, but with partial credit	.943	.949	.926	.023	.940	98.7
7	Less productivity (0.5B)	.642	.660	.650	.018	.650	98.7

Table 1

Some Experimental Results

The model is essentially a dynamic program which calculates an optimal "attention allocation trajectory" for all the blocks present, and then takes the first step of that trajectory. As soon as each new block appears, the dynamic programming calculation is repeated. The model is constrained by three parameters to make it human-like. The parameters may be adjusted according to various criteria until the model best fits experimental data. One parameter is a time delay τ, simply adjusted to match human motor reaction time plus decision time.

A second parameter is a linear discounting of importance of later blocks in various alternative trajectories which the dynamic programming algorithm compares to determine which trajectory costs least. This discount rate we call β. Zero β means that, in present evaluation of alternative trajectories for future action, what the model earns in the more distant future weights just as heavily as what it earns in the very next step. Large β means the model discounts the future completely and only considers alternative next steps.

A third parameter, γ, is a linear discount rate on distance of blocks (tasks) from the deadline, determined anew at each successive model iteration. Zero γ means that, in deciding what to do next, blocks far from the deadline are just as heavily weighted as those close to it (multiplied by the blocks' individual importance). Large γ means the model only attends to what is close to the deadline. It is a "putting out bonfires" strategy.

It may seem at first reading that β and γ mean the same thing, but this is not true, and in fact it was our experiments which led us to see this distinction: this aspect of the model grew out of the research. The point is that time into the future, with respect to alternative sequences of (planned) action, is quite different from opportunity time available. In other words, the task which is far from the deadline can be done first, and the one which is close to the deadline done later. The only absolute constraint, of course, is that no task can be "done" after it crosses the deadline.

5. Results from the Model

We now have experimented with the model itself on various multi-task situations. In those situations cited above where all blocks appear at the outset we have verified, as expected, that zero β and zero γ are best. All information is known from the start, and an optimal trajectory as determined by dynamic programming is optimal in an absolute sense.

Curiously, this is not true of the experiment where blocks appear continually. Let us recall that the dynamic programming algorithm computes an optimal trajectory based on what blocks are in view at the time, then commits itself to the first step of that optimal trajectory. Thus, if there is discounting in "planning time", optimal may be to do a relatively unimportant but about-to-disappear task, since there is just time

then to complete an important task which is the only one available. But, while doing the unimportant task, suppose a new important task appears with the same opportunity time as the other important one. A choice must be made between the two important tasks, since only one task can be attended to at a time; one important task must be lost. Had the model expected the new important task was coming it would have attended first to the available important task, ignoring the unimportant close-to-deadline one, and then had time available for the new important one. Instances of this effect are revealed in simulation runs described below.

Our model runs thus far had been made with varying τ values (reaction times) and <u>either</u> varying β or varying γ. τ values have been matched to average reaction times of experimental subjects on a one-run-at-a-time basis.

We have let the computer compare human DM results with computer results separately on the basis of five different criteria: 1) percent value gained for the given run out of the total possible value obtainable; 2) percentage of all completed tasks independent of duration or importance; 3) percentage of time both model and human subject acted on the same tasks at the same time; 4) squared differences between cumulative value gained by model and human, summed over the entire run; 5) squared differences between incremental value gained by model and human for brief time interval, summed over the entire run.

Figures 3 through 7 show examples of five model runs. Figure 3 is for subject KT for the baseline experimental conditions. Figure 4 is for the same subject for a speed 2.5 times as great as the baseline. Figures 5, 6, and 7 are for three different subjects for a productivity half that of the baseline. On each page are ten plots, each plot representing a series of model runs at different values of β (left column, see abscissa below for value of β) with $\gamma = 0$, or model runs at different values of γ (right column) with $\beta = 0$. Points symbolized by X are model runs. The horizontal lines represent human data for the given experimental condition. Circles are comparisons between human and model. Each row is for measures according to a different criterion, as indicated. Thus all points on any vertical slice represent the same model run. Ordinate values of the performance criteria are shown at the right.

Thus, considering the plots in order from top criterion to bottom, the top one is to be maximized (or matched to the line for best fit to human). The X plot of the second one is to be maximized (or matched to the line for best fit to human); the circles on this plot represent % of tasks which are <u>common</u> to model and human, and are to be maximized. The third plot is to be maximized, the fourth and fifth are to be minimized.

For the first criterion (% value gained) it is evident that the model closely approximates the human, at lower values of β or γ doint slightly better (as one would expect for little or no discount) while at higher values doing slightly worse (where the model is not allowed to "plan ahead", i.e., β is large, or is not allowed to consider blocks

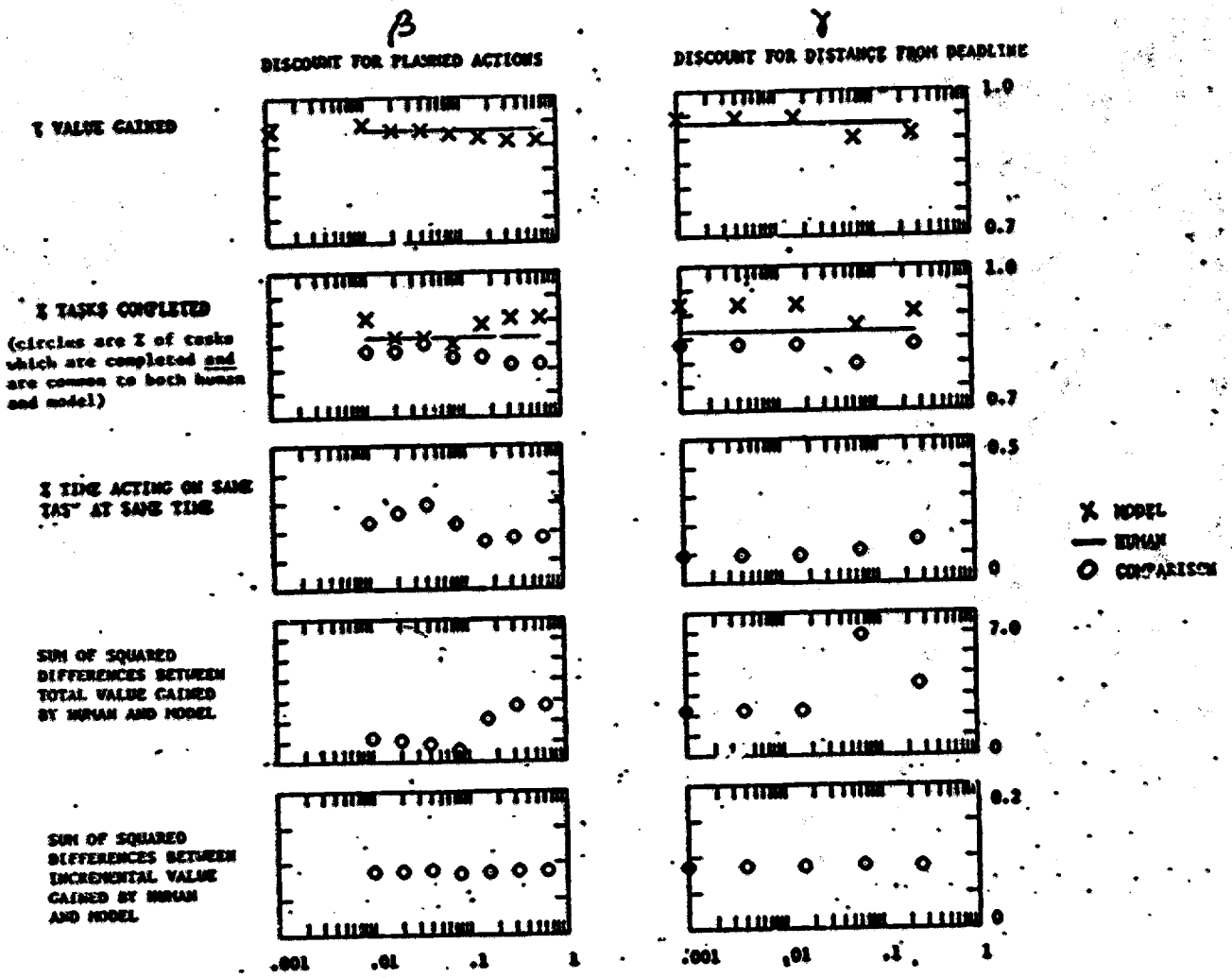

Figure 3. Condition 1. BASELINE:
interarrival 20 sec/queue
speed 0.1 units/sec
productivity 0.5 units/sec
Subject: KT

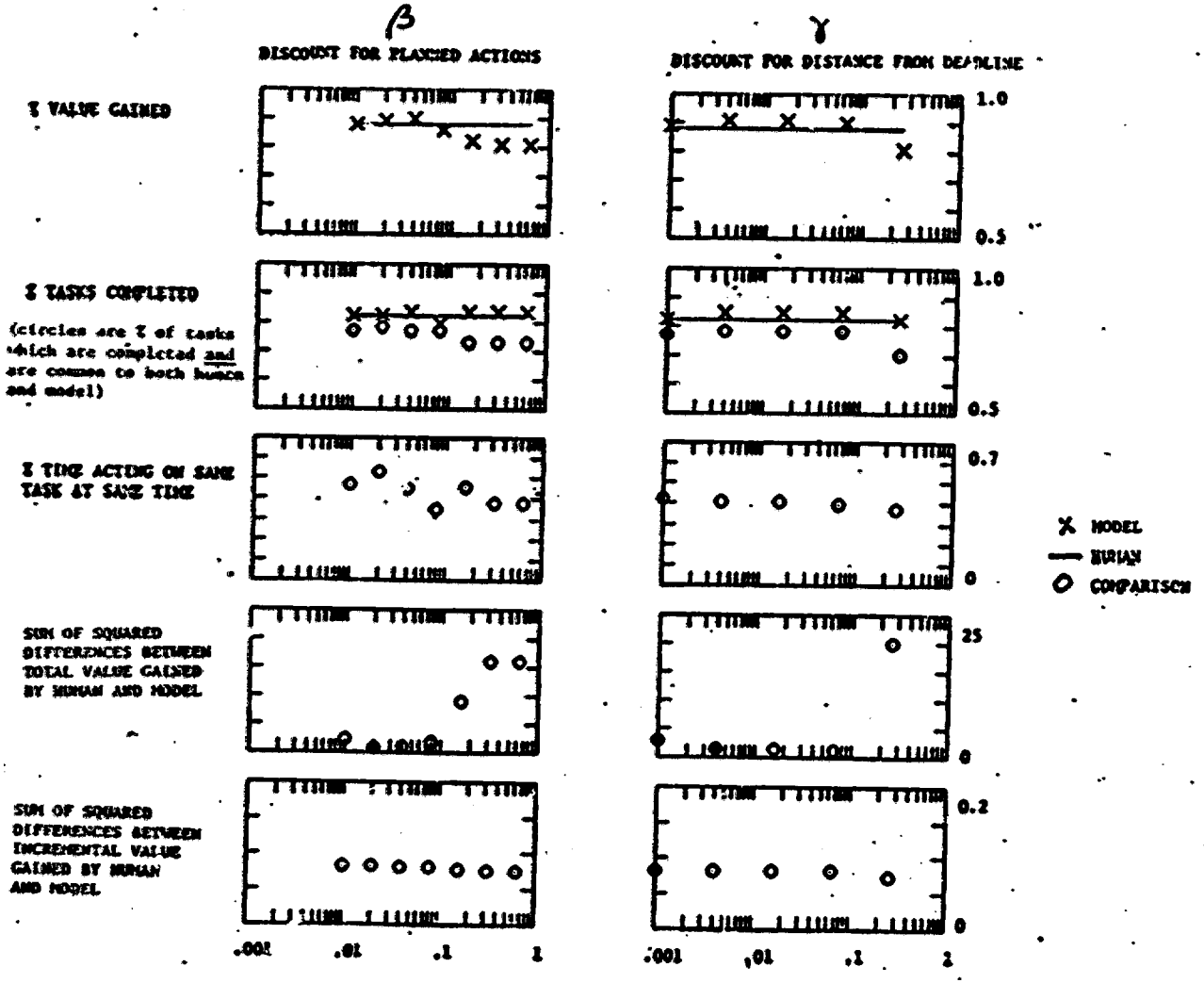

Figure 4. Condition 2
interarrival 20 sec/queue
speed 0.25 units/sec
productivity 0.5 units/sec
Subject: KT

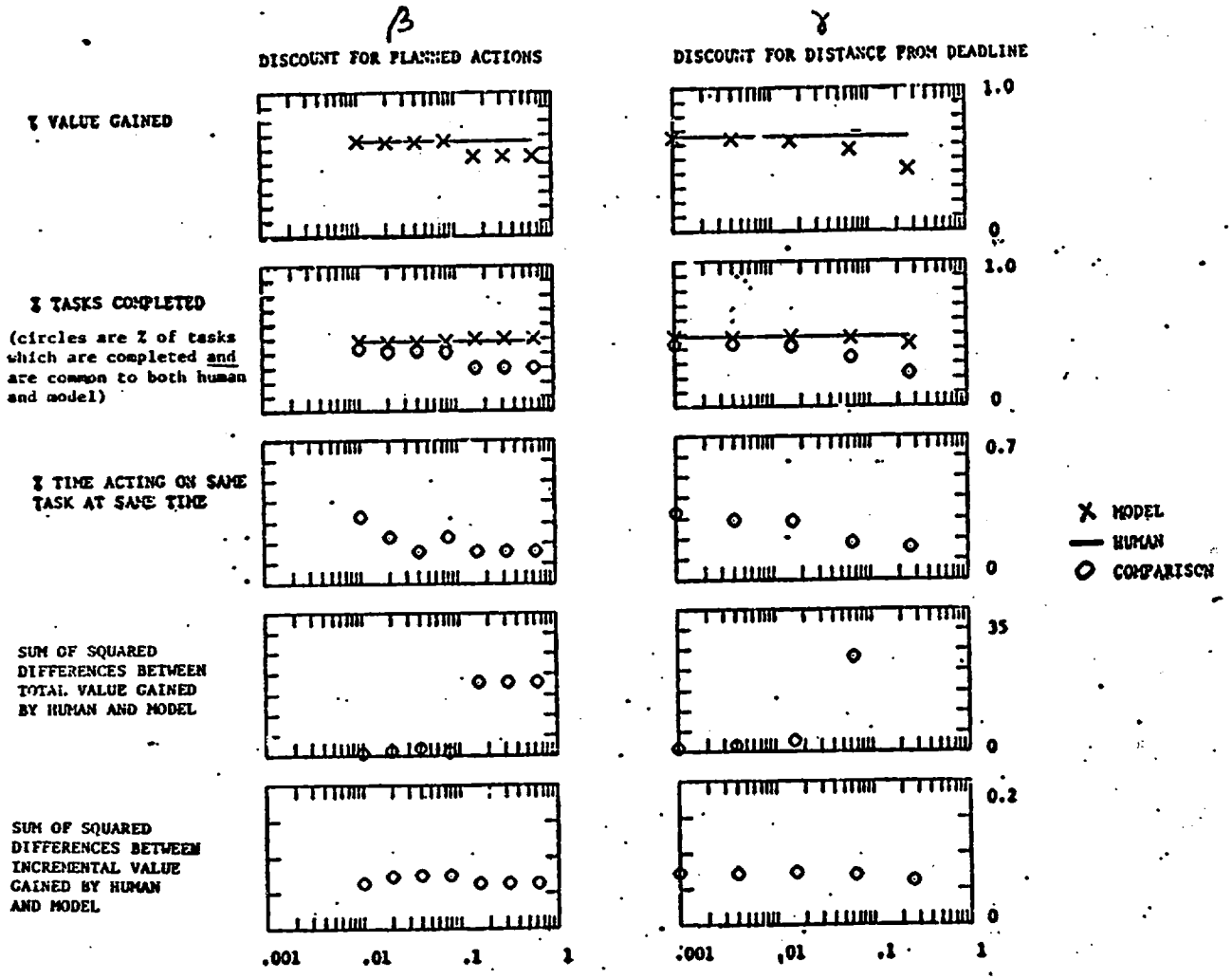

Figure 5. Condition 7
interarrival 20 sec/queue
speed 0.1 units/sec
productivity 0.25 units/sec
Subject: KT

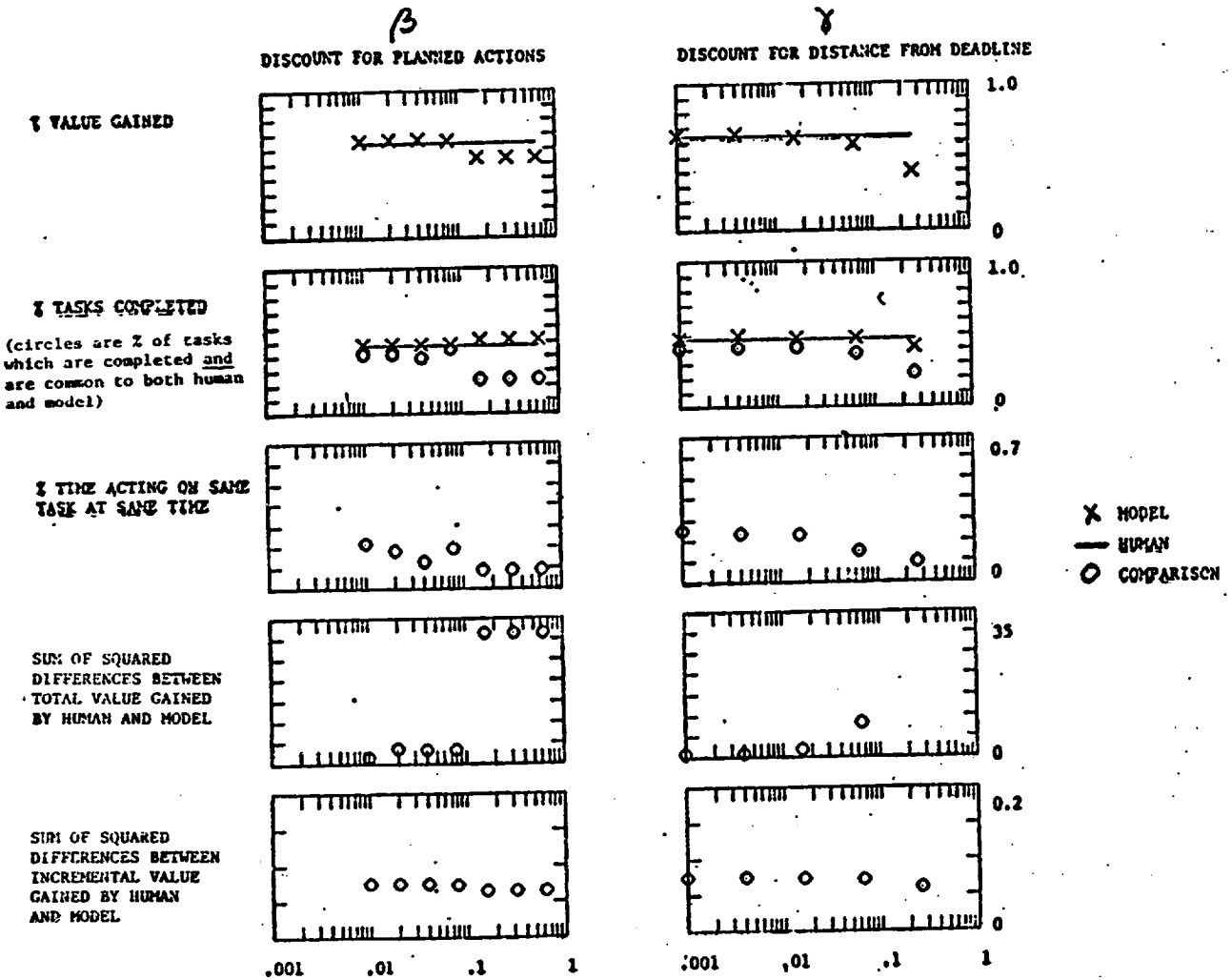

Figure 6. Condition 7.
interarrival 20 sec/queue
speed 0.1 units/sec
productivity 0.25 units/sec
Subject; DY

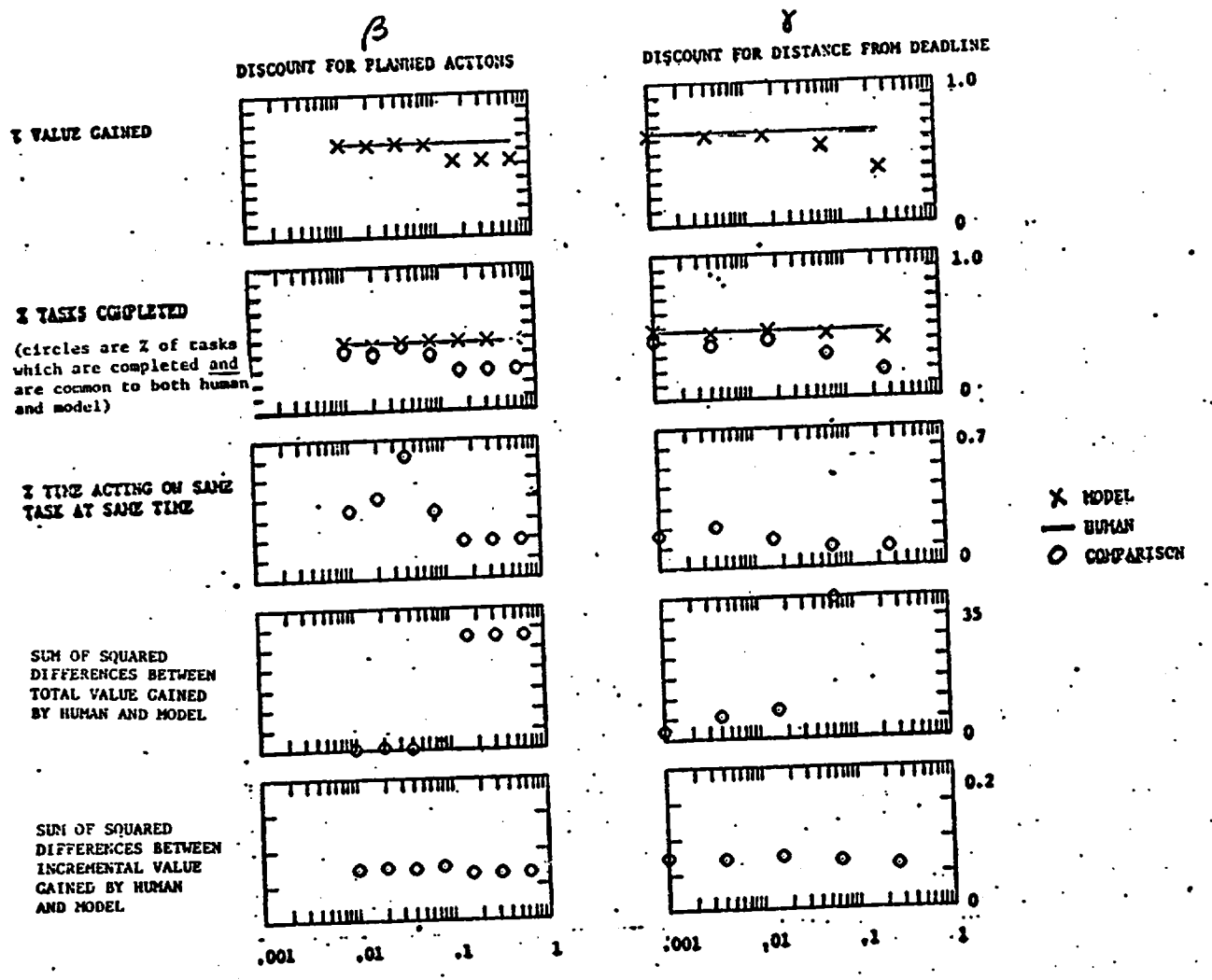

Figure 7. Condition 7.
interarrival 20 sec/queue
speed 0.1 units/sec
productivity 0.25 units/sec
Subject: SJ

far from the deadline, i.e., γ is large). Interestingly, however, for data on the first two pages zero β is not quite as good as a slightly larger β. The theoretical reason for this was discussed above, i.e., with a slight discounting of the future the model is more apt to do the most important block first, and be more open to new blocks which have high payoff.

In everyday terms, this suggests that a person with lots to do, little time to do it, and new tasks continually popping up with relatively short deadlines, should not plan too far ahead. Mostly he should do the most important thing first, ignoring the closest-to-deadline factor. As he has time to see what's coming farther into the future and doesn't expect many new opportunities to be popping up, he should plan ahead.

With respect to the second criterion (% tasks completed) it is interesting that the model and human match precisely in a mid range of β which is also the best match of model to human for tasks which are common to both model and human. This suggests (1) that a β in this range is a good candidate for a model, (2) that the higher task completion capability of the model in other β ranges, without concommitant increase in total value gained, meant it was wasting time on unimportant tasks. The γ fits for this criterion are not so good or so consistent, and we begin to see that γ seems not to be a very meaingful parameter.

As for the next parameter, % of time acting on the same task at the same time, it appears that the β curve peaks at approximately the same value for several of the subjects, but again the γ curve is not very interesting.

The curves for the final two criteria seem to have little to offer, except that the fourth curve consistently takes a jump (gets worse) for β values at 0.1 or larger.

Further experiments will seek to refine the model, the fitting criteria, and possibly add an estimator of future tasks to the optimization algorithm.

Appendix 1. The Model in Relation to Graph Theory*

The paradigm described in the paper will result in a graph $G_T(t) = G(N;A)$ with N nodes and A arcs, where each node represents a task and arcs represent the transfer properties between these tasks. Note that rewards associated with different nodes can be different and delay-(time-) dependent. Also the processing (or service) and availability times of the nodes and the transfer times between them can be different. Therefore in a reward-time (r-t) coordinate framework we have graph $G_T(t)$ as shown in Figure A1.

Note that in Figure A1 τ values represent transfer times between nodes, which incidentally can be direction-dependent, such that precedence constraints can be imposed. t^R, t^D and t^P are "ready-time", "deadline time" and "processing time", respectively. Note that when the rewards associated with the tasks are constant until they hit the deadline, the r-t curve associated with a node will be as shown in Figure-A2a. For the case in which the DM can get partial credit, however, the rewards, rather than being Fixed-Loss, will be as shown in Figure-A2b.

In the Figure A2 t^S is the slack time, i.e. the latest time; if, during which the task is completed all the reward associated with the task can be gained. Note that "time available" is deadline-time minus ready-time:

$$t^A = t^D - t^R.$$

One interesting observation that can be made from Figure-A1 is that in $G_T(t)$ graphs there may not be enough time to get the rewards of all nodes N. In fact, we can infer from the same figure that the best schedule that can be chosen in the particular graph $G_T(t)$ is $\Pi = (2,1,4)$ which does not include node (task) 3.

At this point we digress and consider this sequencing problem in relation to other common combinatorial problems like Job-Shop Scheduling, Traveling Salesperson, etc. (Golden and Magnanti, 1977).

We can differentiate the sequencing problems listed in Table-A1 according to the following criteria:

1) Will multiple journeys between the nodes be counted multiple?
2) Can we add extra nodes?
3) Can the rewards associated with the nodes be delay-dependent?
4) Can the transfer delays between different pairs of nodes be different?
5) Is it imperative to return to the base node?
6) Is it necessary to satisfy the above requirement before a certain delay, T_R?
7) Can the graph G, desribing the problem change dynamically in time?

*See list of symbols at end of Appendix 2.

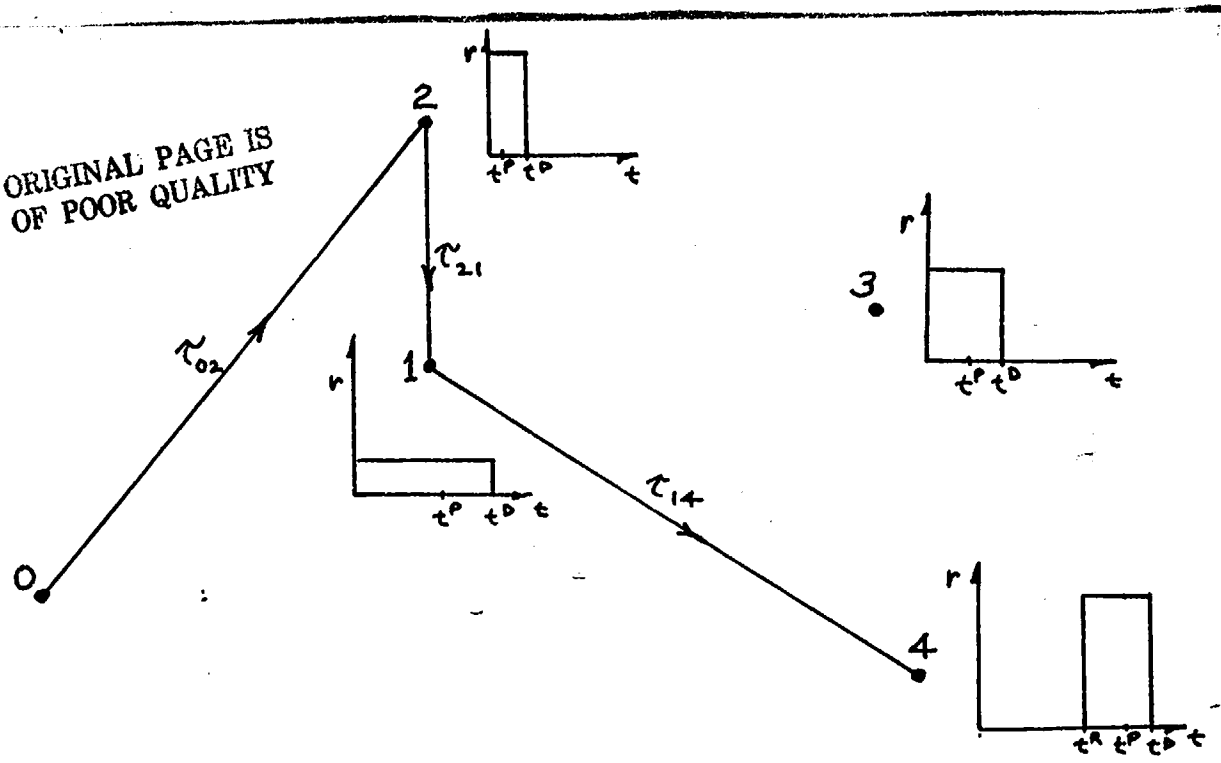

FIGURE A1. A Schedule π = (2,1,4) for Multi-Task Attention Allocation on Graph $G_T(t) = G (N ; A)$.

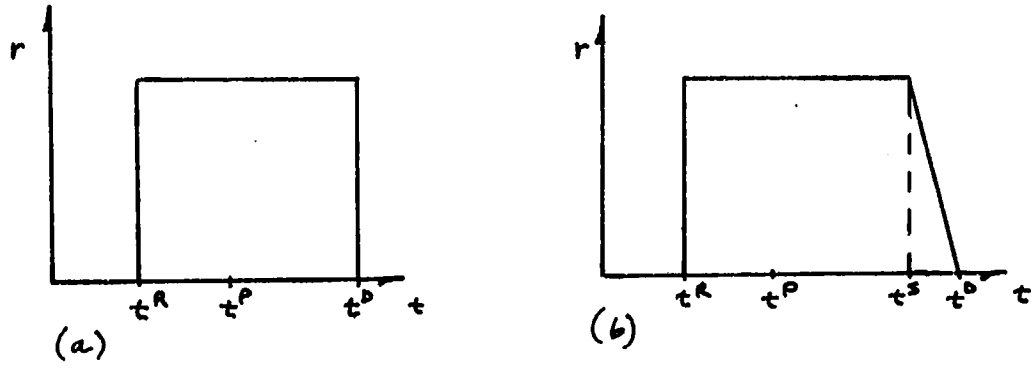

FIGURE A2 Reward-Time Curves for a Task (a) when no Partial Credit is Given, and (b) when Partial Credit is Given.

Using these criteria we have listed some common combinatorial optimization problems with two new ones:

a) Minimum Spanning Trees, MST (Kruskal, 1956)

b) Steiner Tree Problem, STP (Nijenhuis & Wilf, 1975)

c) Job-Shop Scheduling, JSS (Elmaghraby, 1968 and Sahni, 1976)

d) Hamilton Cycle, HC; alias the Traveling Salesperson Problem (Held & Karp, 1962)

e) Open Tulga-Path, OPT; alias Multi-Task Attention Allocation,

f) Closed Tulga-Path, CTP

Note that in Table-A1 the indicator '0' means that the particular criterion need not be satisfied for the problem at hand, while indicator '1' is for the opposite case, with 'N/A' indicating that the criterion is not applicable for the problem. Figure-A3 is a schematic representation of some of the problems. OTP describes Multi-Task Attention Allocation.

Problems vs. Criteria:	MST	STP	JSS	HC	OTP	CTP
1	0	0	1	1	1	1
2	0	1	N/A	N/A	N/A	N/A
3	0	0	1	0	1	1
4	1	1	0	1	1	1
5	0	0	0	1	0	1
6	0	0	0	0	0	1
7	0	0	0	0	1	1

Table-A1. Properties of Various Sequencing Problems.

Before returning to the Multi-Task Supervisory Control, the reader can observe from Figure-A3 that, if the requirement was to serve all the nodes (tasks) with minimum number of controllers (or processors or vehicles or people, etc.) another controller might have been assigned to node-3 in Figure-A3(iii), and the OTP problem will become an advanced version of the 'Bin-Packing Problem'. (Johnson, 1974) The reader may note here the case of computer aiding (2nd. controller) of the human operator (1st. controller). (Rouse, 1977) Similarly in Figure-A3(iv) an extra vehicle can serve node-3 and come back to the base node before T_R; however, unless the return time T_R is sufficiently large, node-4 cannot be served whatever the number of vehicles, but as T_R increases 3, then 2 vehicles will be enough to serve all the nodes: the CTP then becomes an advanced 'Vehicle-Routing Problem". (Golden, 1976)

We can see from Figure-A3 that Multi-Task Attention Allocation Paradigm is representable by the OTP Combinatorial Problem when we consider that node 0 (base node) is where the DM currently is, and 4 tasks

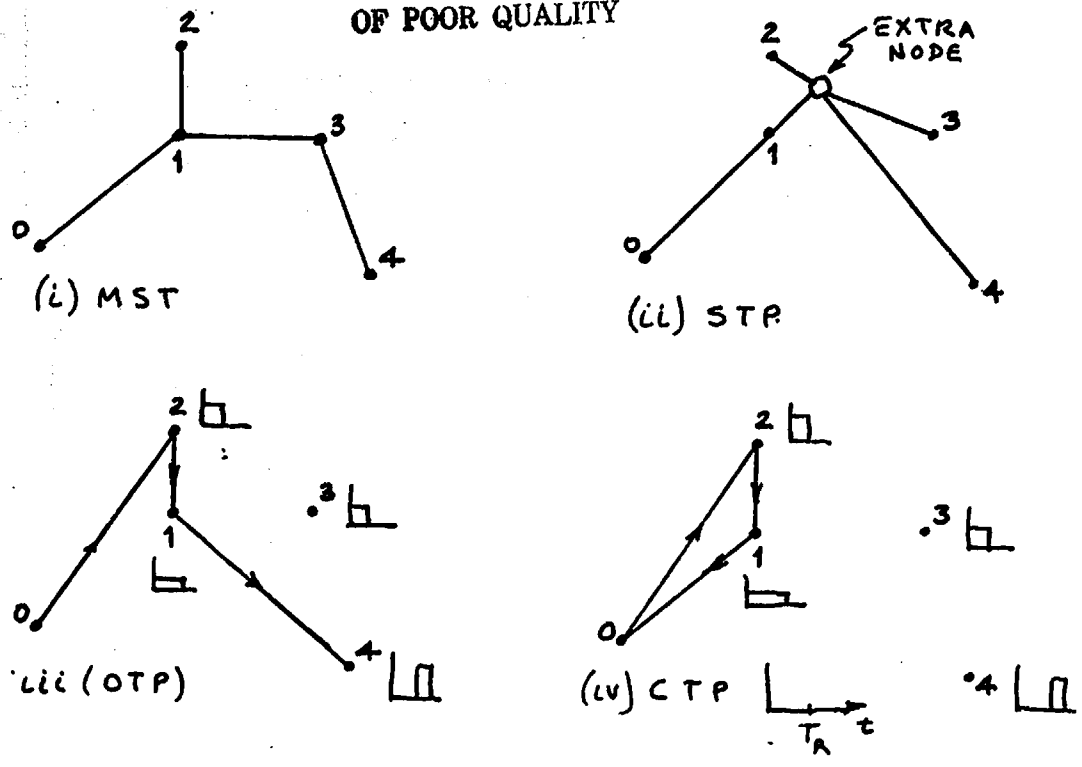

FIGURE A3 Graphical Representation of Some Sequencing Problems

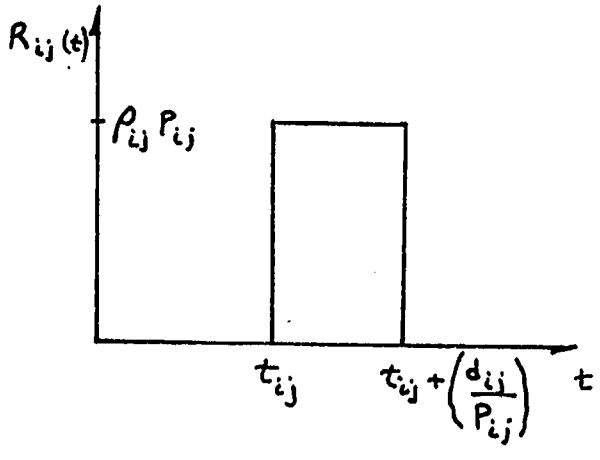

FIGURE A4. The Return, for a Task, as a Function of Time for the Partial Credit Mode

are (or will be) available with different properties. The DM will then act on the first task of the optimal schedule, $\Pi^\circ = (2,1,4)$, i.e. task 2.

Note however, that new tasks may appear on this graph $G_T(t)$ probabilistically according to the interarrival rates and with the task parameters explained in the paradigm section, and the thing to be maximized is the reward gained at the end of the experiment, so that tasks that are going to appear cannot be ignored. That is to say: since graph $G_T(t)$ is time-dependent, then the optimal schedules $\Pi^\circ(t)$ on them are time-dependent too.

Appendix 2. Optimization Algorithm of the Model

In choosing his control, i.e., which task to at upon, we can model the DM as an optimal controller who maximizes his expected returns over a planning horizon. (Koopmans, 1964). In particular, the DM will act to maximize his expected total returns over a finite planning horizon, T, with a discount function $B(\beta, t)$:

$$\max_{\Pi} \quad r(\Pi) = E\left[\int_0^T R_\Pi(t)\,dt \right]$$

where $$R_\Pi(t) = \sum_{(i,j)\in\Pi} R_{ij}(t) \cdot B(\beta, t)$$

in which the summation is over all the tasks (i,j), which collectively make up the ordered task set, schedule Π, that the DM expects to act upon over his planning horizon. $R_{ij}(t)$ is the return he gets for acting on (or completing) the task (i,j) during (or at) time t.

For the case in which the DM gets credit continuously while acting on a task, the $R_{ij}(t)$ will be as shown in Figure-A4.

In Figure-A4, t_{ij}, P_{ij}, d_{ij}, p_{ij} represent the time at which the DM plans to start acting on the task, the value density of the task, the duration of the task, and the productivity of the DM for the task (i,j), respectively.

If however, the DM is going to get (full) credit only after successfully completing a task, then the $R_{ij}(t)$ will be as shown in Figure-A5.

The DM, in effect, will choose at each decision point a schedule $\Pi^\circ = (\Pi_1^\circ, \Pi_2^\circ, \ldots)$ that he intends to act upon to maximize his expected returns, and then he will actually act upon the first task Π_1°, in this ordered set of tasks.

It is probable and acceptable that he might have to give up on acting on some tasks when their 'available times' are small - due to their high speed and/or due to their proximity to the deadline - or when they have comparatively low value densities, especially in compe-

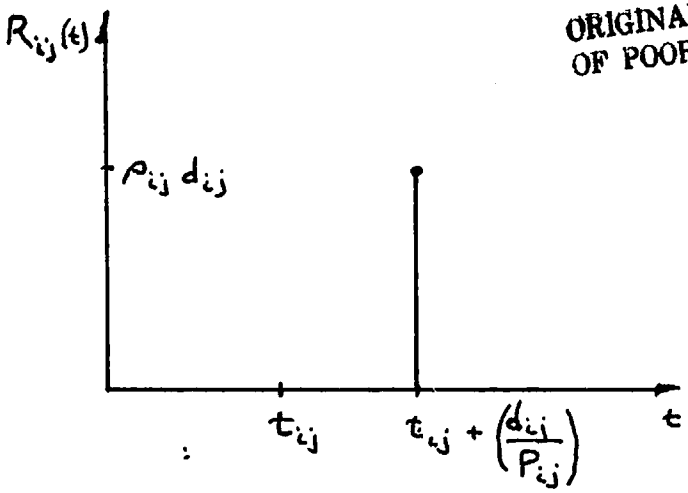

FIGURE A5. The Return, for a Task, as a Function of Time for the No-Partial Credit Mode.

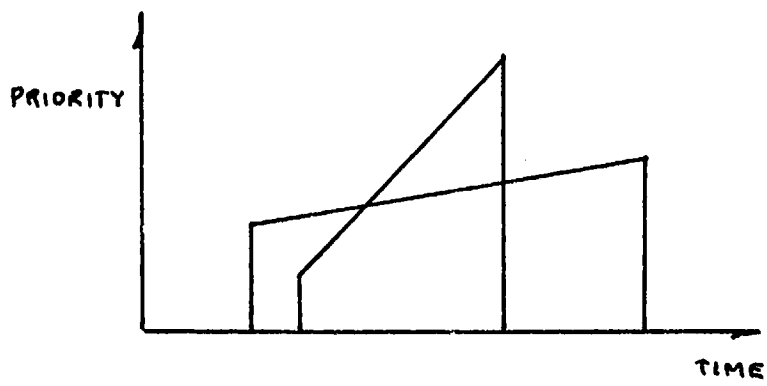

FIGURE A6. Increasing Priorities of Different Tasks as They Wait to be Served.

tition with other simultaneously available tasks which are preferred in these respects. Another important parameter, of course, is the transfer time $\tau_{ii'}$, between the queues. He has to consider the fact that he will end up getting no credit for a period of time when he transfers his control from the i.th queue to the i'.th one.

The algorithm for finding the optimal schedule of tasks Π°, to act upon is:

Algorithm T_PATH

Input (usage, T_R, $\tilde{X}_{ijk}$, $\tau_{ii'}$, B, G)

The input parameter 'usage' indicates whether an OTP or a CTP is desired, and if it is a CTP, T_R is used as the required return time to the base node. τ is the transfer-delay time matrix between the queues of tasks and B, and G are discount functions on future returns and on tasks away from the deadline-tasks with larger slack-times -, respectively.

Note that the system state tensor $\tilde{X}_{ijk}$ specifies the various task parameters for each given instant of time like:

1) whether the task is available (display) or not, L_{ij} (=1 or 0)
2) the return associated with the task as a function of time, $R_{ij}(t)$
3) the processing/service time of the task $t^{P}_{ij}(t)$
4) the 'available time' of the task, $t^{A}_{ij}(t)$

Output optimal schedule Π°, and discounted present value $r(\Pi^{\circ})$ and completion time $c(\Pi^{\circ})$ associated with it.

Step-1 [Initialize]

```
for i = 1 to I do
for j = 1 to J_i do
while L_ij = 1 do /* is the task available ? */
transform (i,j) to ℓ and
generate the tuple (r(ℓ), c(ℓ))

r(ℓ) = R_ℓ(t = 0)
c(ℓ) = τ_0ℓ + t^P_ℓ(t = 0)

end
end
```

Note that $R_{\ell}(t) = \int_{\tau=t}^{\infty} R_{ij}(\tau), B(\beta, \tau) \cdot d\tau$

Furthermore, the tasks currently available are summed to give N, which is also the maximum number of stages, M, the optimal schedule can have.

Step-2 [Generate schedules that are m stages deep]

for m = 2 to M do

generate all m-member-subsets = S

and for each task $\ell \in S$
generate the $(r(\Pi)\ ,\ c(\Pi))$ tuple(s)
-where $\Pi = \{\Pi'\ ,\ \ell\}$, i.e. schedule Π is schedule Π' with task ℓ at stage m.
-for each $\Pi' = \text{order}\{S-\ell\}$, i.e. for each $\Pi' \overline{\equiv} \{S-\ell\}$,
-where the '$\overline{\equiv}$' operator tests whether each member of one set is also contained in the other. (Weinberg, 1971)

$$r(\Pi) = r(\Pi') + R_{\ell}(t = c(\Pi'))$$

$$c(\Pi) = c(\Pi') + \tau_{\ell'\ell} + t^{P}_{\ell}(t = c(\Pi'))$$

where ℓ', is the last task - task at stage (m-1) in schedule Π'.

Eliminate schedules according to the rules:

1) Eliminate the tuples which are infeasible, that is credit cannot be obtained from the last task ℓ in schedule Π before it reaches the deadline; or if usage is CTP, before $(T_R - \tau_{\ell o})$, where $\tau_{\ell o}$ is the transfer time between the queue of task ℓ and the base node 0.

2) Eliminate schedule Π^1, if there is a schedule Π^2, such that:

$$\Pi^1 \overline{\equiv} \Pi^2$$

and $\ell^1 = \ell^2$ or queue of ℓ^1 = queue of ℓ^2

-ℓs are the last tasks -at stage m- in the respective schedules-

and $r(\Pi^1) \leq r(\Pi^2)$

and $c(\Pi^1) \geq c(\Pi^2)$

3) Eliminate the schedules that are less than (m-1) stages deep.

end

Step-3 [Return to the base node if usage is CTP]

if usage = Closed Tulga-Path then
for all schedules Π do

$$r(\Pi) = r(\Pi) + R_{o}(t = c(\Pi))$$

$$c(\Pi) = c(\Pi) + \tau_{\ell o}$$

with ℓ being the last task of schedule Π.

end

Step-4 [Optimal]

The optimal schedule Π° is the one with the property:

$$r(\Pi^\circ) > r(\Pi) \quad \text{for all } \Pi \neq \Pi^\circ.$$

and if

$$r(\Pi^\circ) = r(\Pi) \quad \text{then } c(\Pi^\circ) < c(\Pi)$$

Note that when the rewards of nodes are delay-independent then this algorithm reduces to the dynamic programming formulation of the Traveling Salesperson:Problem. (Held & Karp, 1962). On the other hand, when transfer delays between all tasks are equal and when rewards of all tasks are Fixed-Loss, i.e. constant up to a certain delay (time) and then zero, then the solution will reduce to Job-Shop Scheduling with Deadlines. (Elmaghraby, 1968 and Sahni, 1976).

Several things should be clarified at this point. First, if the model is permitted to get partial credit, as in Figure-A4, then the tasks which will hit the deadline before they can be complted will also be included in the optimization, although with their returns $R_{ij}(t)$ appropriately adjusted to reflect the gain that can be obtained from them before they disappear.

Another point that should be emphasized is that, since all the dynamics of the tasks are known a-priori by the algorithm (and also by the human), there is no need to repeat the optimization unless there is a new task arrival; when no new information is presented, the optimal plan, i.e., the currently optimal schedule will be followed in real time as the tasks in this linked list are completed. It has also been proven theoretically (McNaughton, 1959) that there is nothing to be gained by shifting attention from one task to another and back again, even in the case of no time penalties for doing so. On the other hand, if after a new task arrival the first task in the new optimal schedule is not the task that is currently being attended, then the model will pre-emptively leave the current task to serve the first task in the new optimal schedule. However, the task that was pre-emtively abandoned might still be in the new schedule, and conditions permitting may eventually be re-attended.

The effect of $G(\gamma, t^s)$ will be to adjust the return $R_{ij}(t)$ for acting on task (i,j), by changing the effective value density of the task (i,j) as:

$$R_{ij}(t) = R_{ij}(t) \cdot G(\gamma, t^s_{i,j})$$

where t^s_{ij} is the slack-time of the task, i.e., $t^s = \max_\circ\{0, \lfloor (x/\dot{x}) - (d/p) \rfloor\}$, with $x, \dot{x}, d, p$ representing the current position, speed, current duration and the productivity associated with the particular task, respectively.

Note that the idea of weighing tasks according to their initial priorities plus incremental priority increases as they wait in a queue (Carbonell, 1966 and Jackson, 1965), as shown in Figure-A6, corresponds to the $G(\gamma, t^s)$ function, where the initial priority is determined by the initial proximity of the task to the deadline, and this priority increases as the task approaches the deadline.

It is interesting to note also that, as the speeds of the tasks approach zero, i.e., the deadlines are at infinite future time - and if the transfer times between all the tasks are equal, then the DM is modeled to choose the new task to act upon, according to:

$$\max_{(i,j)} \rho_{ij} p_{ij}$$

This, of course is the familiar result from the Queueing Theory (Smith, 1956) when we consider the productivity of the DM, p_{ij}, as the service rate μ_{ij} and the value density of the task (i,j) ρ_{ij} as the negative cost per unit time delay c_{ij}:

$$\min_{(i,j)} c_{ij}\mu_{ij} \text{ where } c_{ij} < 0$$

List of Symbols

G	graph
t	time
τ	transfer time
$\mathfrak{t}$	dummy time
r	reward available at a node (task)
R	reward gained for a given plan
Π	a schedule — $r(\Pi)$ total discounted return of a schedule; $c(\Pi)$ completion time of a schedule
Π°	that schedule which is optimal
T_R	deadline time for return to base node
T	planning horizon
B	discount function on future returns
β	discount parameter (rate in this case) on future returns
G	urgency discount function
γ	urgency discount parameter (rate in this case)
I	total number of queues
J_i	total number of tasks in queue i.
ℓ	combination of i & j for any task
M	maximum number of stages that optimal schedule can have
m	stage index
d	duration
$\mathfrak{s}$	speed
ρ	value density
p	productivity
x	position

t^P processing time

t^R ready time

t^D deadline time

t^A available time

t^S slack time

References

Elmaghraby, S.E., "The One Machine Sequencing Problem with Delay Costs", J. Ind. Eng., Vol. 19, No. 2, p. 105-108, February 1968.

Golden, B.L. and Magnanti, T.L., "Deterministic Network Optimization: A Bibliography",Networks, Vol. 7, No. 2, p. 149-183, Summer 1977.

Golden, B.L., Ph.D. Dissertation, Sloan School of Management, M.I.T., 1976.

Held, M. and Karp, R.H., "A Dynamic Programming Approach to Sequencing Problems", J. SIAM, Vol. 10, No. 1, p. 196-210, March 1962.

Johnson, D.J., Ph.D. Dissertation, Mathematics Department, M.I.T., 1974.

Koopmans, T.C., "On Flexibility of Future Preference", in Human Judgements and Optimality", (eds. M.G. Shelly II and G.I. Bryan), John Wiley 1964.

Kruskal, J.P., Jr., "On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem", Proc. Amer. Math. Soc., Vol. 7, No. 1, p. 48-50, February 1956.

McNaughton, R., "Scheduling with Deadlines and Loss Functions", Management Sci., Vol. 6, No. 1, p. 1-12, October 1959.

Nijenhuis & Wilf, Combinatorial Algorithms, Academic Press, 1975.

Rouse, W.B., "Human-Computer Interaction in Multitask Situations", IEEE Trans. Syst. Man and Cybern., Vol. SMC-7, No. 5, p. 384-392, May 1977.

Sahni, S.K., "Algorithms for Scheduling Independent Tasks", J.ACM, Vol. 23, No. 1, p: 116-127, Jan. 1976.

Smith, N.E., "Various Optimizers for Single-Stage Production", Naval Res. Logist. Quart., Vol. 3, No. 1, p:59-66, Mar. 1956.

Weinberg, P., "Betriebswirtschaftliche Iogik", Bertelmans Universitaetsverlag* Duesseldorf, 1971.