Vips : a decision support system for visual interactive production scheduling

Document status and date:
Published: 01/01/1988

Document Version:
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

VIPS, A DECISION SUPPORT SYSTEM FOR VISUAL INTER-ACTIVE PRODUCTION SCHEDULING

by

A.W.J. Kolen
A.P. Woerlee   NFI 11.88/05

A.W.J. Kolen
University of Limburg
P.O. Box 616, 6200 MD Maastricht, the Netherlands

A.P. Woerlee
Erasmus University
P.O. Box 1738, 3000 DR Rotterdam, the Netherlands

V_iPS, a Decision Support System for

Visual Interactive Production Scheduling

A.W.J. Kolen
University of Limburg, Maastricht

A.P. Woerlee
Erasmus University Rotterdam, Rotterdam

August 1988

V_iPS is an interactive software package that can be used for production scheduling. It supports the following operations: the construction of assembly line flow-charts, the specification of jobs to be processed, and the search for near optimal schedules. This paper contains a general description of V_iPS, a detailed description of the interface between V_iPS and software in its environment, and a functional description of all available commands.

Contents

Part I
Visual Interactive Production Scheduling

1 Introduction

Many firms have realized that manufacturing can be a competitive weapon. Nowadays manufacturing strategy is an important part of the corporate strategy. The need for both flexible production and the capability to produce orders with tight due dates, has increased the need for a dynamic production scheduling system. Three levels of planning can be distinguished in production scheduling. *Strategic* planning involves decisions about products to sell, capacity, facilities, technology, vertical integration, quality and organization. *Tactical* planning concerns work force sizes and inventory levels. *Operational* planning involves material control and the scheduling of the production system. V_iPS is especially meant for the second part of the operational planning: the scheduling.

Although there is much literature on scheduling topics, the usage of scheduling algorithms in practice is not frequently reported. We believe that algorithms for production scheduling can be introduced in practice. The integration of these algorithms in decision support systems can even increase their usefulness.

Anthonisse et al. [3] argue that a decision support system for planning problems should be able to act as an advisor to the planner (i.e., suggest schedules) but also as an assistant (i.e., evaluate schedules or changes in schedules suggested by the planner). V_iPS satisfies these requirements as it can both be used as an automatic pilot and scratch-pad. V_iPS has a supporting function; the user is in charge and is responsible for the decisions to be made. The use of V_iPS can lead to: fast and simple operational planning, increase of the productive capacity of used machines, reduction of work in process, more flexible scheduling, improved information management, and lesser dependence on the experience and skills of the human planner.

V_iPS also provides some facilities for data entry and report generation. V_iPS does not support the connection with material requirement systems, information systems and data base systems.

2 Problem Type

V_iPS is suited for *generalized job shop* scheduling problems that can be described as follows. A production system consisting of machines and buffers, which provide finite capacity intermediate storage facilities, has to perform a number of jobs. The machines are partitioned in machine groups. A *machine line* is a concatenation of machine groups and buffers. A *machine group* is a collection of unrelated parallel machines, i.e., the processing time of a particular job on this machine group may be different for each of the available machines. A *machine* can process one job at a time and is available in predetermined periods. A machine group can have a preceding buffer to store jobs temporarily. A job in a buffer requires a part

of the buffer's capacity, which may be different for each job. Figure 1 shows an example of a production system.
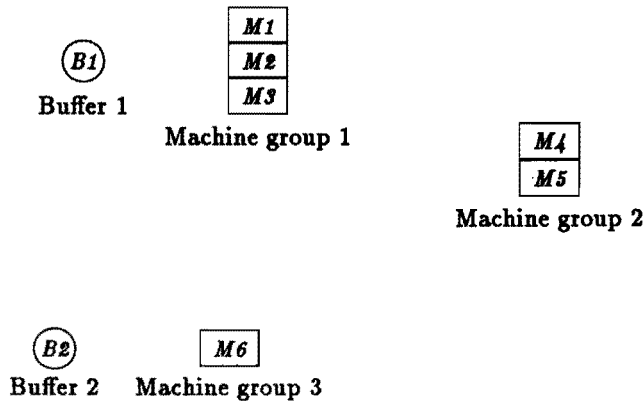
Figure 1: Production system

A *job* consists of a set of operations. For each operation the machine group on which it has to be performed is specified. The order in which the operations have to be processed is given for each job. Therefore a job can be considered as a route through the production system, where a visit to a machine group corresponds to an operation and a visit to a buffer corresponds to temporary storage. Figure 2 shows the route of two different jobs. The numbers indicate the processing time in case of a machine and the capacity requirement in case of a buffer.
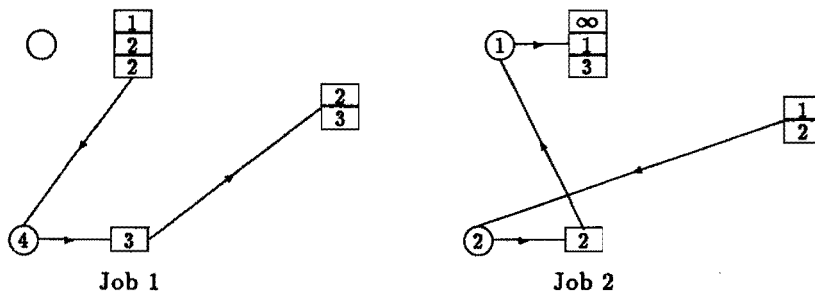
Figure 2: Job routes

Whenever a job is ready to be processed on the next machine group and there is no machine available at that time, the job is transferred to the buffer preceding that machine group. If the buffer is filled to capacity the job has to remain on the last used machine, thereby blocking this machine. The time required to transfer jobs between machines and

buffers is the *transport time*. The *product size* of a job is the number of identical jobs we have to process during the planning session. In some production systems a number of identical jobs is always processed together on a machine group in multiples of a given size, the *transport size*. The transport size is defined per operation and is the minimum allowed length of a sequence of identical jobs processed on the associated machine group. A *task* is a realization of one of those sequences in the generated schedule. Each job has a release date, a due date and a priority. The priority shows the importance that a job is finished before its due date. Each machine has sequence dependent *setup times*, i.e., the time required to adjust a machine for a job is dependent on the job last processed on this machine.

3 Solution Method

Practical scheduling problems are enormous complex and extremely difficult to solve. In solving these difficult scheduling problems, the decomposition approach, where the problem is partitioned into subproblems, seems a natural one. most cases even these subproblems remain hard to solve to optimality but they are much easier to handle. The solution method in $V_i PS$ is based on the above decomposition idea. The machine groups are scheduled successively, one at a time.

Among all unscheduled machine groups a bottleneck machine group is identified with respect to the violation of the due dates. This is the machine group which is scheduled next. Scheduled machine groups provide release and due dates for the operations on unscheduled machine groups. The solution method separates the problem into interrelated machine group problems. The schedules of machine groups are used to reschedule the machine groups until an acceptable solution is found. The planner can influence the progress of the procedure by fixing the start and completion times of particular operations. A decomposition method for the standard job shop problem is described by Adams et al. [2].

4 User Interface

The purpose of a decision support system is to improve the quality of decision making in terms of effectiveness and efficiency. Therefore the system has to deal as effectively as possible with both the task at hand and the specific requirements of the decision maker. As a consequence the user interface of a decision support system is as important as the algorithms.

Another crucial aspect of a decision support system is the representation of the information. In order to construct or change a schedule manually, the user needs an informative representation of the schedule and an easy way to accomplish the desired actions. Therefore a graphical user interface has been chosen. There are two graphical representations implemented in $V_i PS$, a Gantt-chart and a flow-chart. An alphanumerical representation shows the data of machines and jobs in a tabular fashion.

To provide an easy to use control mechanism, a mouse device is used for the interaction between the planner and $V_i PS$. The command structure is menu based. This has the ad-

vantage that it prevents the user from giving illegal commands, and that at all times the planner has a complete view of all available commands.

The menus in V_iPS are so called drop down menus. Moving the mouse cursor to the name of a menu instructs the system to show the menu. There are six menus available. The first menu is the **COMMUNICATION** menu, containing commands to read or write problem instances from or to a file or a data base, or to restart or stop a planning session. The **DESIGN** menu contains commands to define and change a production system. The **SCHEDULE** menu contains commands to construct (partial) schedules for the specified production system. The **REPORT** menu offers the user commands to print information about the problem instance and the constructed schedule. The commands in the **REP-RESENTATION** menu offer the user the possibility to switch to another representation, to adjust the planning horizon or to change the color of a job. With the commands in the **SELECT** menu the planner can make a specific selection out of the available jobs and machines.

If, in spite of all precautions, an error occurs, V_iPS displays an appropriate explanatory message. To obtain more information on the commands the user can invoke the on-line help facility.

If V_iPS requires alphanumerical data from the user, it pops up a dialog window in which all the needed information is described. When constructing a schedule, V_iPS is using Gantt-charts to represent the generated schedules. Infeasibilities in the constructed schedule are displayed in a contrasting color.

5 Graphics Environment Manager

The Graphics Environment Manager (GEM) system has been used to build the user interface of V_iPS. GEM [1,4] is a graphical windowing system which supplies a wide range of graphic routines for drawing boxes, lines, etc. Besides these standard routines GEM also provides facilities for window and dialog handling.

6 Input

In this section, the input specifications for the components of the production system and the jobs are roughly sketched. Data can be retrieved via a data base or can be read from import files, which are specified in more detail in *Part III*. The recurrent pattern mentioned in the off-line period of a machine describes whether an off-line time will occur only once or is returning each day, week, etc.

Machine lines

- name;
- machine groups, buffers.

Machine groups

- name;
- machines.

Machines

- name;
- number of off-line periods;

  per off-line period:

    − start and end time, reason of unavailability, recurrent pattern.

Buffers

- name;
- capacity.

Jobs

- name;
- product size, release date, due date, priority;
- number of nodes in the route graph;

  per machine group node:

    − node identification, processing time per machine, transport size, number of transport batches;
    − preceding nodes in the route graph;

  per buffer node:

    − node identification, required capacity;
    − preceding nodes in the route graph;

- number of arcs in the route graph;

  per arc:

    − transportation time.

Setup times

- machine;

  per setup time:

    − two jobs, setup time.

7 Output

In this section, the information concerning jobs, machines, buffers and schedules is summarized. A detailed description of the format of these files can be found in *Part III*.

Summary jobs

- name of job;
- sum due date violations;
- tasks.

Summary machines

- name of machine;
- total processing, idle, setup, repair time;
- tasks.

Summary buffers

- name of buffer;
- time of change in intermediate storage, storage level.

Schedules

- identification of task, name machine, name job;
- start and completion time of task;
- length of processed sequence.

8 Current Implementation

V;PS has been written in the C programming language and interfaced with the Graphics Environment Manager (GEM). This implies that *V;PS* can run on all fully IBM-compatible personal computers with a high resolution color graphics adapter and monitor. For the development of *V;PS* the following configuration was used: Olivetti M28 (AT compatible computer), Enhanced Graphics Color board (EGC), Logitech mouse, 640 Kb internal memory and 20 Mb external storage. Without any adjustment *V;PS* can also be used in coherence with the Enhanced Graphics Adapter (EGA).

Part II
Functional Description

1 User Interface

A new problem instance can be constructed with the commands in the **DESIGN** menu or an existing problem instance can be retrieved from an external file or a data base with commands in the **COMMUNICATION** menu.

A mouse device is used for the interaction of the user with V_iPS. A command is executed by moving the mouse cursor to the desired command and clicking the left mouse button once. Two other mouse techniques are used. First, the double click method. Its purpose is to get information about jobs, machines, etc. To use this method, the mouse cursor should be moved to the desired item on the screen, and the left mouse button should be pressed twice without moving the cursor. Secondly the click-drag method. Its purpose is to position a task, job, machine or buffer somewhere on the screen. To use this method, the mouse cursor should be positioned on the item which should be moved. Next, the left mouse button should be pressed and not be released, the mouse cursor moved to the desired position on the screen and finally the mouse button released.

All available commands will be described in detail in the following sections using the structure of the menus. The three dots succeeding some of the commands indicate that V_iPS will ask the user to supply additional input.

2 Communication

This menu contains the commands to read and write a problem instance, to restart a scheduling session and to cancel an executed command.

Retrieve...

input: *Inquired: database.*

effect: V_iPS opens the specified database and retrieves the machine and job data. The current problem is removed from memory.

Store...

input: *Inquired: database.*

effect: V_iPS opens the specified database and the the machine and job data are stored in the specified data base.

Import...

input:	*Inquired: import file.*
effect:	The machine, job, setup and schedule data from the specified import file are read.
remark:	See *Part III* for a description of the import file format.

Export...

input:	*Inquired: export file.*
effect:	The machine, job, setup and schedule data are written to the specified export file.
remark:	See *Part III* for a description of the export file format.

Restart

effect:	The current schedule is removed and the user can start the construction of a new schedule.

Undo

effect:	The changes to the schedule due to the last command given are canceled.

Quit

effect:	End the planning session.

3   Design

This menu contains the commands to design a new problem instance or alter an existing problem instance. The current problem instance is displayed on the screen. Another problem instance can be selected by using the commands in the **COMMUNICATION** menu. To start the design of a problem instance the user should select one of the commands in the **DESIGN** menu. These commands cause a switch to a flow-chart representation. The right hand side of the screen is the worksheet displaying the layout of the machines and buffers. The left hand side of the screen displays three icons, representing a machine, a buffer and a job. A layout of a production system is constructed by dragging the desired icons to a position on the screen. After a machine or buffer has been placed somewhere in the worksheet, V_jPS will open a dialog window in which all necessary information about the machine or buffer, should be entered by the user.

After the layout of the production system has been completed the jobs have to be specified. A new job is defined by selecting the job icon at the left hand side of the screen. V_jPS will display a dialog window in which the user has to specify some characteristics of the job.

The route of a job starts at a receiving area and ends at a shipping area. An arc in the route graph is indicated by clicking on two consecutive machines or buffers. Since the route

graph is directed, the order in which these machines and buffers are selected is extremely important.

Construct

effect: The screen is divided in two parts. The left part shows the icons of a machine, a buffer, a receiving area, a shipping area and a job. The right hand side displays the current problem instance in a flow-chart.

Delete

input: Job or machine or buffer.

effect: The selected item is deleted from the problem instance.

remark: When a machine or buffer is deleted, all the jobs using the resource will be listed together with their predecessors and successors in the route graph. The user is obliged to adjust the route graph of the listed jobs.

Group

input: Machine(s).

effect: The specified machines are arranged in a machine group. The machines are selected by pressing the left mouse button on a desired machine. By pressing the left mouse button again on a selected machine, this machine will be deselected.

Ungroup

input: Machine group.

effect: The specified machine group is decomposed. The machine group is selected by pressing the left mouse button on one of the machines in the group.

Information

input: Job or machine or buffer.

effect: All available information about the selected item is displayed.

remark: It is possible to alter the listed information. This can be useful if the processing time of an operation has to be changed, or the time in which a machine is available.

Retrieve...

input: *Inquired: data base.*

effect: The problem instance in the specified database will be retrieved and placed in memory thereby erasing the old one.

Store...

input: *Inquired: data base.*

effect: *VjPS* stores the problem instance in the specified data base. The data will remain in memory for additional manipulations.

4 Schedule

This menu contains all the commands necessary to construct a schedule either by hand or by using the available scheduling routines, and the commands to alter an existing (partial) schedule. The commands in the **SCHEDULE** menu are divided in three groups, commands related to tasks and job manipulations, commands for automatic scheduling purposes, and commands for data manipulations. The manipulation of tasks can have an enormous impact on the other tasks in the schedule. Moving a single task can make the schedule infeasible. The manipulation of jobs and the automatic pilot functions do not change the start and completion times of the tasks which are already scheduled.

To start the creation of a schedule the user should select one of the commands in the **SCHEDULE** menu. The schedule will be displayed as a Gantt-chart. At the left hand side of the screen the jobs are displayed in a list, sorted in order of increasing due dates. The remaining part of the screen is reserved for the Gantt-chart. After the selecting of a command for the manipulation of tasks *VjPS* splits the left hand side of the screen in two parts. The upper region is used to display the jobs and the lower portion is used to display the operations of the selected job. Clicking on the button in the left top corner of the bottom window will remove this window. Infeasibilities in the schedule caused by modifications will appear in a red color in the Gantt-chart.

Add

input: Job or operation or task.

effect: The specified job or operation is added to the schedule. It is also possible to move a task in the schedule to another position.

Delete

input:	Job or operation or task.
effect:	The specified job, operation or task is removed from the schedule.
remark:	V_jPS will display an error message if the user tries to delete a locked task. The removal of a task can make the schedule infeasible.

Interchange

input:	Two tasks.
effect:	The first task is moved to the position of the second task in the schedule. Likewise the second task is moved to the position of the first task.
remark:	If the interval is too small for a task V_jPS will push up the tasks after the tasks which are interchanged. If the user tries to interchange a task which cannot be performed on the machine of the other task, or tries to interchange a locked task V_jPS will display an error message.

Split

input:	Task.
effect:	The specified task will be split in two parts.
remark:	V_jPS seeks a multiple of a transport size which is as close as possible to the position in the task where the mouse button was pressed. If the task cannot be split V_jPS will display an error message.

Join

input:	Two tasks.
effect:	Both tasks should correspond to the same operation. The two tasks will be joined at the start time of the first one.
remark:	If the interval is too small to join the tasks V_jPS will push up the tasks after the first task.

(Un)lock

input:	Job or operation or task.
effect:	The start and completion time of the specified job, operation or task will be fixed or freed.

Patch

effect: The current schedule is validated and corrected. First, the validation concerns two aspects: the detection of infeasibilities in the start and completion times of operations and the violation of restrictions, such as periods reserved for machine maintenance. Secondly V_jPS will try to remove infeasibilities in the schedule by means of a simple correction method.

Create

effect: The unscheduled tasks of the jobs are placed in the schedule.

Improve

effect: V_jPS will seek a better schedule of all tasks in the current schedule which are not locked.

Retrieve...

input: *Inquired: data base, schedule.*

effect: The specified schedule will be retrieved in memory erasing the old one provided that the problem instance of the specified schedule matches the one in memory.

Store...

input: *Inquired: data base, schedule.*

effect: The schedule will be stored in the database.

5   Report

This menu contains the commands to print information about a given problem instance. Reports are generated on file and should be sent to the printer after the termination of V_jPS by using the OUTPUT application of GEM.

Gantt-chart

effect: A Gantt-chart of the current schedule is generated.

Machine...

input: *Inquired: machine.*

effect: A table of the start and completion times and size of the tasks performed on the specified machine is generated.

Buffer...

input: *Inquired: buffer.*

effect: The usage of the buffer is reported in a tabular manner.

Job...

input: *Inquired: job.*

effect: The tasks of the specified job are reported in a tabular manner.

Schedule

effect: The schedule is summarized in separate tables per job. The table includes the names of the machines and the tasks executed on the machines with their start and completion times.

Facility

effect: A summary report about the production system is created.

6 Representation

This menu contains the commands to select a problem representation. The Gantt-chart representation displays the machine schedules via a drawing of the start and completion times of the tasks. The flow-chart view displays the problem instance as a map. The table view displays a list of start and completion times of tasks per machine.

Gantt-Chart

effect: Transition from the current problem representation to the one oriented towards time.

Table...

input: *Inquired: machine.*

effect: Transition from the current problem representation to the alphanumeric one oriented towards time.

Flow-Chart

effect: Transition from the current problem representation to the one oriented towards machines and buffers.

Color Job...

input: Job; *Inquired: color.*

effect: The color of the specified job is changed.

Zoom...

input: *Inquired: scaling factor or planning interval.*

effect: The user can define a scaling factor of the time axis and alter the planning period. *VjPS* will adjust the Gantt-chart using the specified scaling factor or planning interval.

7   Select

This menu contains the commands to select a subsection of the problem.

Period...

input: *Inquired: planning interval.*

effect: The specified time interval is selected. Only jobs with a release date before the end of the interval and a due date after the start of the interval will be shown.

Line...

input: *Inquired: machine line(s).*

effect: The specified machine line is selected. Only these machine lines will appear in the Gantt-chart and flow-chart.

Group...

input: *Inquired: machine group(s).*

effect: The specified machine groups are selected. Only these machine groups will appear in the Gantt-chart and flow-chart.

Machine...

input: *Inquired: machine(s).*

effect: The specified machines are selected. Only these machines will appear in the Gantt-chart or flow-chart.

Buffer...

input: *Inquired: buffer(s).*

effect: The specified buffers are selected. Only these buffers will appear in the Gantt-chart and flow-chart.

Job...

input: _Inquired: job(s)._

effect: The specified jobs are selected. Only these jobs will be shown in the list
                at the left hand side of the screen. Algorithms are only applied to the
                jobs in the selected group.

remark: This command has no effect on the jobs which are already in the schedule,
                but not in the selected group.

Part III
Input/Output Specification

	# positions
node type	1
production system:	
system name	24
per machine line:	
line name	24
# nodes	2
per node:	
node type	1
node name	24
per machine group:	
group name	24
# parallel machines	2
per machine:	
machine name	24
per machine:	
machine name	24
# off-line periods	2
per off-line period:	
start time	12
end time	12
reason of unavailability	1
recurrent pattern	1
per buffer:	
buffer name	24
capacity	6

Table 1: Production System Format Specification

			# positions
per job:			
job name			24
product size			6
release date			12
due date			12
priority			2
# nodes in route graph			2
per node:			
	node number		2
	node type		1
If machine or machine group:			
	transport size		6
	# transport batches		6
	# available machines		2
	per machine:		
		machine name	24
		processing time	8
If buffer or buffer group:			
	# available buffers		2
	per buffer:		
		buffer name	24
		required capacity	6
	# preceding nodes		2
	per preceding node:		
		node number	2
# arcs in route graph			2
per arc:			
	transportation time		8

Table 2: Job Format Specification

	# positions
per machine:	
machine name	24
per setup time:	
1st job name	24
2nd job name	24
setup time from 1st to 2nd job	8

Table 3: Setup Times Format Specification

		# positions
per job:		
job name		24
$\sum$ due date violations		8
# tasks		4
per task:		
	machine name	24
	start time	12
	completion time	12
per machine:		
machine name		24
$\sum$ processing times		8
$\sum$ idle times		8
$\sum$ setup times		8
$\sum$ repair times		8
# tasks		4
per task:		
	job name	24
	start time	12
	completion time	12
per buffer:		
buffer name		24
# changes in storage		4
per change:		
	time	12
	storage level	6

Table 4: Summery Format Specification

	# positions
per task:	
machine name	24
job name	24
start time	12
completion time	12
length of processed sequence	6

Table 5: Schedule Format Specification

Remarks:

- For each *"per"* and *"if"* the user should continue the data specification at a new line.

- Code of *node type* in table 1: 1 = machine, 2 = buffer, 3 = machine group, 5 = machine line, 6 = production system.

- Code of *reason of unavailability* in table 1: 1 = off-line, 2 = repair.

- Code of *recurrent pattern* in table 1: 1 = daily, 2 = weekly, 3 = monthly.

- The *node name* in table 1 is the name of a machine, machine group or buffer.

- The *node number* in table 1 and 2 is a unique number identifying a particular node in the route graph.

- Dates and times of length 12 are stored in the following format: day (1-2), month (3-4), year (5-8), hour (9-10), minute (11-12).

- Times of length 8 are expressed in seconds.

References

[1] *GEM Programmer's Guide*. Digital Research, 2nd edition, 1987.

[2] Joseph Adams, Egon Balas, and Daniel Zawack. The shifting bottleneck procedure for job shop scheduling. *Management Science*, 34(3):391–401, March 1988.

[3] J.M. Anthonisse, J.K. Lenstra, and M.W.P. Savelsbergh. *Functional Description of CAR, an Interactive System for, Computer Aided Routing*. Report OS-R8716, Centre for Mathematics and Computer Science, Amsterdam, The Netherlands, September 1987.

[4] Phillip Balma and William Fitler. *Programmer's guide to GEM*. Sybex Inc., Berkeley, CA 94710, USA, 1986. ISBN 0-89588-297-3.