An advisor module for tactical and strategic state-oriented planning systems with an application in manpower planning
Boers, H.; Venema, W.Z.; Wessels, J.

Published: 01/01/1992

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 author's 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

Citation for published version (APA):
Memorandum COSOR 92-52

An Advisor Module for Tactical and Strategic State-Oriented Planning Systems with an Application in Manpower Planning

H. Boers,
W.Z. Venema,
J. Wessels

Eindhoven, December 1992
The Netherlands
An Advisor Module for Tactical and Strategic State-Oriented Planning Systems with an Application in Manpower Planning.

H. Boers¹, W.Z. Venema¹, J. Wessels¹,²)

¹) Eindhoven University of Technology
Faculty of Mathematics and Computing Science
P.O. Box 513, 5600MB Eindhoven, The Netherlands

²) International Institute for
Applied Systems Analysis
A-2361, Laxenburg, Austria

Abstract

A user-controlled breadth-first search method is discussed for generating qualitative and quantitative advice in tactical and strategic state-oriented planning situations that are modelled with a network structure. An application within the Manpower Planning System FORMASY is demonstrated.

Keywords: Decision support systems, Manpower planning, Push models, Pull models.
1 Introduction

Tactical and strategic planning problems can be very complicated, even if one manages to convert them satisfactorily to a mathematical model. Complications can arise because of the long-term impact of decisions and because of the delay in the effects of these decisions.

Many processes can be modelled with a network structure or, more specifically, with a directed graph. In such a graph, the nodes describe the state of the process according to a number of qualitative and quantitative characteristics. The arrows of the graph describe the dynamics of the process. We will assume that the time characteristic of nodes connected by an arrow differs by one unit of time. Thus, identical process states at different points in time are represented by distinct nodes.

The arrows could denote physical flows, as commonly found in logistics and manpower planning. They could also denote a more general control mechanism, for instance one that describes the application of a special treatment, or the spread of knowledge or disease. One could argue that a control mechanism always involves a transfer of something, and that it therefore describes a flow as well; however, this would be a flow on which no strict conservation law can be applied (conservation law: for all (sets of) nodes the in-flow equals the out-flow).

A complete description of the behaviour of a process requires a complete description of its constituent nodes and of its dynamics. In the case of planning problems the dynamics is based on a current policy. We assume the availability of this information and of a mechanism that allows a planner to modify the dynamics. This means that the decision variables are parameters connected to the arrows, and that the current values of these parameters are known. Thus, the evolution of the modelled process is determined by an initial state and by a policy. The evolution mechanism, however, need not be deterministic but may be based on probabilities.

Now, we can classify a state-oriented planning process as the following sequence of planning actions:
• A planning object is chosen, i.e. a set of network nodes is selected.

• The (forecasted) state of the planning object is analysed. The (forecasted) state of the planning object is compared with user-defined goals, so that possible discrepancies can be detected.

• Planning measures are generated that could reduce discrepancies, i.e. (aggregate) arrows are identified where parameters can be modified in order to achieve an improved policy with respect to the planning object. Of course, a planner will prefer some domain-specific vocabulary rather than stating his or her measures in terms of nodes and arrows.

Although the generated measures are in the first place qualitative, it is desirable that the effect of a change to parameter settings can be calculated or estimated, in order to assess whether a measure contributes substantially, and to decide whether additional measures are needed.

Under conditions that will be described in the next section, AI techniques can be used for generating useful planning measures with quantitative effects. A general breath-first search method with user-controlled locality is discussed in section 2 and a practical application in the field of Manpower Planning is demonstrated in section 3.

2 A breath-first search method with user-controlled locality

2.1 A first outline

Consider a well-defined network structure, and assume that the planning object is a well-described compact subset of the network model. For example: all persons in grade X with sex *female* and ages between 20 and 40. We will call such rectangular, compact, areas of the network “blocks”. Thus, the planning object is a block and we will call it DESTINATION. This name
denotes the state-orientation of the planning process in contrast to a dynamic pattern-orientation.

With the selection of DESTINATION the planner has not only defined a part of the process he or she is interested in, but also an aggregation level which guides the entire planning process. Planning measures should refer to this aggregation level: they should incorporate about the same amount of detail as was involved in the definition of DESTINATION.

The aggregation level that was determined by the selection of DESTINATION also affects the view of the dynamics of the process. We assume that each arrow can be referred to with a term that denotes its special relevance to the planner. For instance, in manpower planning some of the arrows could be referred to as promotion transitions; in logistics as deliveries; and in health service planning as treatments. In aggregated views, the arrows that make up an aggregated arrow share the same planner reference term.

As mentioned before, arrow descriptions contain the parameters which may serve as decision variables. Such parameters could be promotion fractions in manpower planning, tonnages in logistics and doses of medication in health service planning. Of course, not all arrows need to have parameters. For instance, manpower planning retirement flows are mostly autonomous.

We now propose the following outline of a breath-first search method for generating qualitative planning measures:

1. Analysis stage:

   - The (predicted) state of DESTINATION is generated from the model data, i.e. from the initial state and a given policy.
   - The (predicted) state of DESTINATION is compared with user-defined goals, so that possible discrepancies can be detected.

2. Measure-generating stage:

   - Aggregate arrows are generated from a source block (SOURCE) to a transitional block (THROUGH), such that there is a route of
aggregate arrows from the transitional block \textit{THROUGH} to the planning object \textit{DESTINATION}. The transitional blocks are generated in order of increasing route length. In this way a recursion mechanism is defined that creates, for each transitional block, both the aggregate arrow(s) and the source block(s) simultaneously (cf. figures 1 and 2).

- For each source block \textit{SOURCE} with transitional block \textit{THROUGH} and the connecting aggregate arrow, the proposal to change parameter settings in the subject arrows is converted to a planning measure in a planner language, taking into account the level of detail associated with \textit{SOURCE}, \textit{THROUGH} and the aggregate arrow.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{figure1.png}
\caption{Schematic view of a simple 1-period measure generator.}
\end{figure}

- In addition to the qualitative measure generator, we assume that it is feasible to calculate or estimate the effect of a generated measure. This is no strong restriction, since we already assumed that the (forecasted) state of the process evolves mechanically from an initial state and a policy. So an evaluation function, defined on the instances of the state of \textit{DESTINATION}, and a forecast generator will suffice.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{figure2.png}
\caption{Schematic view of a simple multi-period measure generator.}
\end{figure}
2.2 The measure generator

The 'block'-wise approach outlined above suggested a simple recursion mechanism. However, the following complications might occur:

- The aggregate arrows are not really transitive: when an aggregate arrow from a source block SOURCE to a transitional block THROUGH is linked to a route of length $N-1$ of aggregate arrows from THROUGH to DESTINATION, the resulting route of length $N$ may be interrupted in THROUGH. Interruptions could emerge as a consequence of neglecting the internal structure of a transitional block, e.g. different parameters of the aggregate arrow in view could apply to different subsets of the transitional block.

One might ignore this problem arguing that the calculator module, which calculates (or estimates) the ultimate effect of planning measures, will filter all planning measures associated with interrupted routes, before the conversion to the planner's vocabulary takes place: only those planning measures will be proposed that contribute substantially to a better performance.

A more efficient algorithmic behaviour may be obtained by narrowing the aggregate arrows in a route of length $N-1$ from a transitional block THROUGH to the object block DESTINATION such that the extended route from SOURCE to DESTINATION is not, or less likely, to be interrupted. This leads to the generation of planning measures on a more detailed level. That does not need to be a disadvantage, since the details that ultimately show up in the proposed measures are relevant to the planning process. Besides, one might consider to always generalize detailed planning measures to a higher level of abstraction (post-processing).

- The recursion mechanism might yield identical or nested planning measures because different routes of aggregate arrows might yield source blocks that are identical or nested (nested meaning here: subsets of each other).

The appropriate way to overcome this problem is to perform some post-processing: For all integers $N$ the generated planning measures
associated with a route of length \( N \) are gathered in a list. This allows us to eliminate duplicates or nestings before conversion to planning measures in the planner's vocabulary.

Once all planning measures associated with a certain route length have been presented to the planner, it is useful to include a query whether or not the program should generate planning measures that take one additional time unit to take effect. This allows the planner to control the locality of the solutions the measure generator comes up with.

- The calculator module, which computes the quantitative effects of the proposed planning measures, cannot be incorporated into the measure generator.

When the arrows denote flows and the parameters denote quantities of flow, one might be tempted to conceive the route of aggregate flows from a source block to the object block as a black box with some inflow-outflow ratio, and to try to modify the inflow-outflow ratio when the black box is extended with yet another aggregate flow. However, there may be many more different competing routes of flows from a node within the source block to a node within the planning object than the one that is included in a particular route of aggregate flows. Hence, the gain (or loss) of flow derived from increasing (or decreasing) the quantity of flow from a source block to a transitional block may be reduced, or in the worst case become negative, by the simultaneous reduction (or increase) of a competing flow.

Thus, in general, the calculator module must be devised in a more global manner, such as with a forecast generator and an evaluation function as suggested in section 2.1.

For many applications the method outlined above is too simple: in general it is not possible to change one arrow's parameter setting without having to change parameter settings for other arrows. For instance, arrows could denote flows subjected to a conservation law. We will therefore refine the method by generating combinations of aggregate arrows when the aggregate arrow from a source block to a transitional block is not separately adjustable. In that case, the proposed measure, expressed in the planner's vocabulary, will be based on a change to the shared parameter.
In the situation sketched in figure 3, the block size of DESTINATION determines the block size of SOURCE. The latter in turn determines the block size of OTHER and the distinction of the aggregate arrow-to and the aggregate arrow-from. The recursion mechanism can still be applied (cf. figure 4).

The method is suited for a wide variety of planning situations, the only major restriction being the locality of the parameters. Their adjustments are interpreted as planning measures: we generate (combinations of) aggregate arrows, and hence limited parts of the network, in order to focus on the subject parameters in isolation of the rest of the network.

As we will see later on, however, the method is also useful when parameters that apply to different periods are subject to an overall constraint. For example, only gradual changes to the dynamics of a process may be permitted (consistency of policy).
2.3 The analysis stage

In the analysis stage the (forecasted) state of the planning object is compared with user-defined goals. We assume that the mechanical generation of the (forecasted) state of the planning object is relatively easy and fast. The need for this assumption follows from the fact that the calculator module operates in a global manner: it generates, for each modified policy, a different instance of the state of DESTINATION that has to be compared with the user-defined goals. Thus, the generation of the (forecasted) state of DESTINATION is repeated often.

Because the comparison with user-defined goals is repeated equally often, we suggest that the analysis stage starts with converting the goals relevant to the planning object to one weighted goal, in the form of an arithmetic expression that is defined on all possible instances of the state of DESTINATION.

The conversion can be a difficult and time-consuming process. Goals that apply to different aspects of the planning object have to be weighted, but it is also necessary to link subgoals, for different sets of nodes within the planning object, to an overall goal for the entire planning object. The latter task may lead to exhausting combinatorial problems if the subgoals may be defined on overlapping sets of nodes. However, the conversion procedure has to be performed only once, so it is legitimate to invest a relatively large amount of effort and CPU time in it.

2.4 Modification of parameter settings

The measure generator in the first place generates aggregate arrows or combinations of aggregate arrows. Then, a modification of subject parameters is interpreted as a planning measure.

In many applications it is obvious in which direction a parameter could be adjusted to arrive at an improved policy. For instance, if the arrows denote flows and the analysis stage pointed to a shortage of goods in DESTINATION it is clear that sensible measures correspond with an increase of the quantity
of flow from a source block to a transitional block, perhaps in conjunction with a decrease of the quantity of flow for competing aggregate flows.

This is an example of how domain-specific knowledge can be incorporated into the measure generator in order to determine the direction and the magnitude of a parameter adjustment rather than trying a whole series of adjustments.

Often, there are overall constraints on the parameter values, for instance for assuring a stable policy. If constraints are hard, that is, violation is not permitted, they can be incorporated into the measure generator by including a procedure for determining the amount of freedom for the parameter in view.

If some of the constraints are soft, that is, if violation is permitted but at a cost, optimization methods can be used to derive an optimal policy. In that case, the method outlined so far can be used to let a planner - who may know little or nothing about mathematics and optimization methods - establish a (preferably small) set of parameters that could serve as degrees of freedom for an optimization method. A practical example will be given in Section 3.

2.5 Locality control

When a planning measure is adopted to obtain an improved policy with respect to the (forecasted) state of the planning object, it is generally desirable that only a limited part of the process will be affected. Besides, the effect of a temporary change of policy often dilutes as time goes by.

Therefore, it is preferable to search for planning measures that affect only parts of the network in the near vicinity of the planning object. This is the reason why we devised the measure generator as a reiterated breath-first search method rather than as a reiterated depth-first search method, and it is also why we included into the generator a mechanism to let the user decide whether or not to generate routes of larger length.
3 An application in Manpower Planning

3.1 Introduction

In Manpower Planning, the goal of the decision making process is to match the future personnel availability with the expected future requirement. An extensive description of this problem area can be found in [1] and [3]. We will demonstrate an application of the method, outlined in section 2, to support a planner in the search for medium- or long-term planning measures. Our approach would fit in a set-up for decision support in manpower policy making as described in [2].

We will restrict ourselves to organizations that can be modelled with a fixed network structure. We assume that the employees are classified according to a number of qualitative and quantitative characteristics, such as grade, grade age, age and sex. Besides a distribution of the personnel over categories, transitions between categories and transitions between a category and the outside world are of interest. Thus, model properties have to be specified such as the transition possibilities between categories, the promotion fractions, wastage fractions, retirement and early-retirement ages, early-retirement fractions, recruitment possibilities and recruitment numbers. We assume the availability of the current values of the transition fractions and the recruitment numbers, which determine the current policy for a number of years (the planning period). These values may vary from year to year.

Furthermore, we assume that the (expected) manpower requirement is known, i.e. that the planner has defined occupation goals a priori, or that he or she is able to judge during a planning session whether a forecasted occupation points to a surplus or shortage of personnel. In section 3.2 we will discuss an advisor program that assists a planner with the selection of the appropriate decision variables, i.e. the recruitment numbers and transition fractions, that may serve as degrees of freedom for reducing the forecasted discrepancy between the manpower availability and requirement. The advisor is an interactive program and the interaction takes place in a manpower planning vocabulary.
If a planner adopts a qualitative planning measure proposed by this advisor program, the concerned degrees of freedom could be passed to an overall optimization method, taking into account other occupation goals as well as constraints on transitions and recruitment number. Thus, the underlying mathematical model could be a mixed push-pull model.

3.2 The advisor program

The advisor program is devised as a tool to assist the user with the choice of the appropriate degrees of freedom (parameter changes) to change a current policy into one that matches the user's goals. Of course, the user does not need to be aware of the underlying mathematical model or of degrees of freedom. To the planner, the program presents manpower planning measures that correspond to the appropriate degrees of freedom, rather than the degrees of freedom themselves. It operates as follows.

We assume that the user has specified a well-defined class of employees by selecting one year in the planning period and a range of values for the other characteristics (such as grade, grade age, age and sex).

The reason for allowing ranges of values for characteristics rather than demanding a choice between a single value and all possible values (projection) is that some characteristics can have a special relevance in combination with a range of values of another characteristic. For instance, the question "how many women occupy executive positions in the company in the year X ?" is of interest if the organization considers positive discrimination actions as a valid instrument.

Assume therefore that the class of employees of interest is specified by a completely instantiated list of eight parameters. In the following we shall call this class of employees DESTINATION and we will make no distinction between the class of employees and its specifying list.

Hence, with DESTINATION = [y, g, g1, ga, ga1, a, a1, s] the class of employees in the year y, with grades in the range g..g1, with grade ages (or grade seniority) in the range ga..ga1, with ages in the range a..a1 and sex s ('male', 'female' or 'any') is specified a priori.
Once the program is activated with a specification of DESTINATION, the (forecasted) actual occupation number of DESTINATION is computed and, if possible, a target value for that occupation number is derived from explicitly stated goals. The user is informed about the actual occupation number and, if possible, about the associated target value and the way the latter is derived ('explicit', 'topdown', 'bottom-up', 'mixed'). When no target value can be derived from already available data, the user is asked whether the actual occupation number points to a surplus or shortage of personnel.

When the actual and target numbers differ significantly, or when the user states that the actual number points to a surplus or shortage, an inference engine is started for generating qualitative planning measures that diminish the discrepancy with respect to the occupation in DESTINATION.

The measures are generated according to a breath-first search algorithm with respect to the reversed time at which the measures can be taken. First, planning measures are generated that can be taken in the year $y - 1$, then those that can be taken in the preceding year and so on until the first year of the planning period is reached.

When the amount of planning measures that can be taken in a certain year gets exhausted, the user is asked whether the measure generator should continue. In this way the user controls the locality of the considered measures.

3.2.1 The measure generator

Figure 5 gives a schematic view of the measure generator when the method of section 2 is applied.

The generator begins by generating a route of aggregate flows from a transitional block to DESTINATION, thereby creating instantiations of the transitional block THROUGH. This process can be described with a relation $route(I, THROUGH, DESTINATION)$ where $I$ denotes the length of the route.

For each instantiation of THROUGH, aggregate flows from a source block to THROUGH are generated, thereby creating different instantiations of the source block SOURCE. If this aggregate flow is not separately adjustable, a
Figure 5: a schematic view of the measure generator.
competing aggregate flow from SOURCE to another block OTHER is generated, too. This process can be described with the following relations: recruitment(SOURCE, THROUGH), promotionto(SOURCE, THROUGH), promotionfrom(SOURCE, OTHER), earlyretirement(SOURCE, OTHER), stayto(SOURCE, THROUGH), and stayfrom(SOURCE, OTHER).

The predicates stayto and stayfrom denote residence flows, i.e. flows describing merely a shift of age and grade age. The meaning of the other predicates should be clear from their names.

The relations promotionto/2 and promotionfrom/2 are not completely trivial because we will allow a great variety of aggregation possibilities. Production rules for these relations have to be formulated separately.

Table 1 gives sensible default choices for the ranges for the source blocks with respect to the characteristics grade and grade age for different types of instantiations of THROUGH in the definition of the relation promotionto(SOURCE, THROUGH). A similar table could be formulated for the relation promotionfrom/2.

For specific organizations, different or supplementary production rules can be defined for the relations promotionto/2 and promotionfrom/2. This could be useful if there is a clustering with respect to the characteristics grade and grade age that has a special relevance (education, departments, etc.).

The route from THROUGH to DESTINATION is made up of promotion and residence flows. These flows are more detailed than the flows defined with the relations promotionto/2 and stayto/2 because we try to avoid interrupted routes from SOURCE to DESTINATION. Therefore these promotion flows are defined with a relation precisepromotionto(T1, T2) which differs from the relation promotionto(T1, T2) in the additional restriction that the current policy prescribes a non-zero transition fraction to T2 both for the subset of T1 with the lowest grade age number and for the subset of T1 with the highest grade age number. Furthermore, a residence flow from T1 to T2 can only be part of a route from THROUGH to DESTINATION if the current policy prescribes that not all employees in T1 retire, early-retire or are promoted. This additional restriction can be denoted with the relation somestay(T1).
<table>
<thead>
<tr>
<th>THROUGH</th>
<th>one single grade age</th>
<th>range of grade ages, but not all</th>
<th>any grade age</th>
</tr>
</thead>
<tbody>
<tr>
<td>one single grade</td>
<td>another grade, one single grade age</td>
<td>another grade, any grade age</td>
<td>another grade, any grade age</td>
</tr>
<tr>
<td>range of grades, but not all</td>
<td>a not included grade, any grade age</td>
<td>a not included grade, any grade age</td>
<td>a not included grade, any grade age</td>
</tr>
<tr>
<td>any grade</td>
<td>(no promotion)</td>
<td>(no promotion)</td>
<td>(no promotion)</td>
</tr>
</tbody>
</table>

Table 1: default distinguished “promotionto” flows.

A knowledge representation for generating the planning measures, before conversion to natural language, could look like this:

**Initialisation**:

DESTINATION = [Y,G1,G2,GA1,GA2,A1,A2,S].

currentyear(T).

**production rules**:

successor(1).
successor(I) if successor(I-1).
route(0,DESTINATION,DESTINATION).
route(I,SOURCE,DESTINATION)
    if route(I-1,THROUGH,DESTINATION)
    and stayto(SOURCE,THROUGH)
and somestay(SOURCE).
route(I, SOURCE, DESTINATION)
  if route(I-1, THROUGH, DESTINATION)
  and precisepromotionto(SOURCE, THROUGH).
degreeoffreedom(I, DESTINATION, MEASURE)
  if \( I > 0 \) and \( I \leq Y - T \)
  and route(I-1, THROUGH, DESTINATION)
  and recruitment(SOURCE, THROUGH)
  and MEASURE = [recruitment, THROUGH].
degreeoffreedom(I, DESTINATION, MEASURE)
  if \( I > 0 \) and \( I \leq Y - T \)
  and route(I-1, THROUGH, DESTINATION)
  and promotionto(SOURCE, THROUGH)
  and stayfrom(SOURCE, OTHER)
  and MEASURE = [prom, SOURCE, THROUGH, stay, OTHER].
degreeoffreedom(I, DESTINATION, MEASURE)
  if \( I > 0 \) and \( I \leq Y - T \)
  and route(I-1, THROUGH, DESTINATION)
  and promotionto(SOURCE, THROUGH)
  and not stayfrom(SOURCE, OTHER)
  and promotionfrom(SOURCE, OTHER)
  and MEASURE = [prom, SOURCE, THROUGH, prom, OTHER].
degreeoffreedom(I, DESTINATION, MEASURE)
  if \( I > 0 \) and \( I \leq Y - T \)
  and route(I-1, THROUGH, DESTINATION)
  and stayto(SOURCE, THROUGH)
  and promotionfrom(SOURCE, OTHER)
  and MEASURE = [stay, SOURCE, THROUGH, prom, OTHER].
degreeoffreedom(I, DESTINATION, MEASURE)
  if \( I > 0 \) and \( I \leq Y - T \)
  and route(I-1, THROUGH, DESTINATION)
  and stayto(SOURCE, THROUGH)
  and earlyretirement(SOURCE, OTHER)
  and MEASURE = [stay, SOURCE, THROUGH, early_retirement].

measure generator : 
successor(I) and degreeoffreedom(I, DESTINATION, MEASURE).
Variables are denoted with capitals or the underscore character (_). The measures are generated as instances of the list \( MEASURE \), and can be converted to the planner's vocabulary, usually in terms of (partially) exchanging the flow between \( SOURCE \) and \( THROUGH \) with the flow between \( SOURCE \) and \( OTHER \). In which direction the exchange has to take place depends on the question whether there is an overoccupation in \( DESTINATION \) or an underoccupation.

A relation \( \text{translate}(SIGN,MEASURE) \) can be devised for the translation of \( (SIGN,MEASURE) \) to a proposal to the planner in natural language if the list \( MEASURE \) has been generated by \( \text{degreeoffreedom} \) and \( SIGN \) has been instantiated in the analysis stage to 'surplus' or 'shortage'. Such a proposal could look like “\( \text{You could increase the promotion chances in 1993 for employees in grade 4 with grade age 6 to grade age 0 of grade 5.} \)”.

3.2.2 Computing the maximum contribution of measures

Each time a qualitative measure is generated with \( \text{degreeoffreedom} \) and translated to the planner's vocabulary with \( \text{translate} \), the maximum contribution of this measure to the occupation number in \( DESTINATION \) is computed. This process is structured as follows.

- An exact forecast is made for determining the number of employees to whom a proposed planning measure applies, and who reach \( DESTINATION \), assuming that the measure would be applied to the maximum extent.

- A similar forecast is made to determine that number when the policy remains unchanged.

- By subtracting the latter number from the former, the maximum contribution of the measure is obtained (negative results denote a decrease of the occupation number, positive results denote an increase).

Note that the whole process of generating planning measures with the advisor program also could be devised to generate the corresponding degrees of freedom for an optimization method. Due to interference from competing goals
and numerous constraints, the effect of a qualitative measure that was generated by the advisor program may turn out to be smaller than the computed maximum contribution suggested. Moreover, the optimization method may produce a policy in which the proposed measure is not applied to the maximum extent. Nevertheless, computed maximum contributions indicate which of the proposed qualitative planning measures can contribute substantially to the occupation goal of interest.

3.2.3 The analysis stage

As mentioned before, the inference engine that generates qualitative planning measures is started whenever a significant surplus or shortage of personnel in DESTINATION is predicted.

Since the user may define occupation goals at various levels of aggregation, a procedure is needed to derive an implicit occupation target value for DESTINATION from those explicitly-defined occupation goals. In other words, we need a procedure for the conversion of the set of all user-defined occupation goals for blocks that have overlap with DESTINATION to a target occupation number for DESTINATION. When it is not possible to derive such a target occupation number for DESTINATION the planner will have to decide whether the forecasted occupation number in DESTINATION points to a surplus or shortage of personnel.

Let $B_1, B_2, ..., B_n$ be the blocks that have overlap with DESTINATION and for which the user has defined explicit target occupation values: $T_1, T_2, ..., T_n$ respectively. A target occupation number for DESTINATION can be derived if there is a conjunction of disjoint elements from the set $\{B_1, B_2, ..., B_n\}$ that covers DESTINATION entirely. An explicit derivation is obtained if one of the elements of this set covers DESTINATION exactly. A bottom-up derivation can be obtained from a conjunction of disjoint elements of $\{B_1, B_2, ..., B_n\}$ consisting of more than one element if that conjunction covers DESTINATION exactly. A topdown derivation can be obtained if one of the elements of $\{B_1, B_2, ..., B_n\}$ covers DESTINATION, but not exactly. Finally, a mixed derivation can be obtained from a conjunction of disjoint
elements of \{B_1, B_2, \ldots, B_n\} consisting of more than one element if that conjunction covers \textit{DESTINATION}, but not exactly.

Target occupation values for coverings are obtained by adding the target values of their constituent elements. A target occupation number for \textit{DESTINATION} can then be estimated from the target occupation number for a non-exact covering by scaling according to their initial occupations. Hence, \textit{topdown} and \textit{mixed} derivations do not express explicitly-specified goals, and it is therefore desirable that a planner can overrule the results from such derivations.

One sensible approach to derive a (possibly implicit) target occupation number for \textit{DESTINATION} is to determine the smallest (in terms of the number of included network nodes) conjunction of disjoint elements of \{B_1, B_2, \ldots, B_n\} that covers \textit{DESTINATION} entirely. This, however, is a tough combinatorial exercise. In section 2.3 we mentioned that it is worthwhile to invest a lot of effort in the conversion procedure for deriving one weighted goal defined on all instances of \textit{DESTINATION}. General algorithms for exactly solving combinatorial problems tend to require extreme amounts of computing time. Therefore, we propose the following procedure for solving the combinatorial problem sketched above.

\textbf{stage 1:} Split the set \{B_1, B_2, \ldots, B_n\} into the sets \textit{UNIQUES} and \textit{CHOICES}, such that \textit{UNIQUES} contains the elements that are disjoint with all other elements.

\textbf{stage 2:} Generate conjunctions of disjoint elements of \textit{CHOICES} that cannot be extended with another element of \textit{CHOICES}. Join each conjunction with the set \textit{UNIQUES} to a set \textit{CANDIDATES}.

\textbf{stage 3:} Test for each instance of \textit{CANDIDATES} whether or not it covers \textit{DESTINATION} entirely.

\textbf{stage 4:} Select the smallest (in terms of the number of included network nodes) element from the set \textit{CANDIDATES} that passes the test of stage 3.

20
Let \( \text{MAX} \) be an upper bound on the number of instances of \( \text{CANDIDATES} \) that is allowed to fail the test of stage 3. If this bound is exceeded, we conclude that combinatorial explosion forces us to renounce the idea of exactly determining the smallest conjunction of disjoint elements of \( \{B_1, B_2, ..., B_n\} \) that covers \( \text{DESTINATION} \) entirely. Then we can pursue a more rough course, for instance by merely combining projections, i.e. elements that specify for each characteristic either an atom or the maximum range.

The conjunctions generator of stage 2 generates conjunctions in decreasing lexicographical order (with each conjunction we can associate an \( n \)-dimensional binary vector whose \( i \)-th component is 1 if \( B_i \) is included in the conjunction and 0 otherwise; thus a lexicographical ordering of conjunctions is defined). Thus the generator will not generate different permutations of the same conjunction. Moreover, the conjunctions generator can be implemented rather efficiently with the following recursion mechanism:

- Select an element \( \text{FIRST} \) of \( \text{CHOICES} \) to be the first to be included in the conjunction.
- Create the subset \( \text{FIRST_TAIL} \) of elements of \( \text{CHOICES} \) that appear behind \( \text{FIRST} \).
- Remove from \( \text{FIRST_TAIL} \) all elements that are not disjoint with \( \text{FIRST} \). This yields a set \( \text{REMAINED} \).
- Split the set \( \text{REMAINED} \) into the sets \( \text{REMAIN_UNIQUE} \) and \( \text{REMAIN_CHOICES} \) such that \( \text{REMAIN_UNIQUE} \) contains the elements of \( \text{REMAINED} \) that are disjoint with all other elements of \( \text{REMAINED} \).
- Generate conjunctions of disjoint elements of \( \text{REMAIN_CHOICES} \) that cannot be extended with another disjoint element from the set \( \text{REMAIN_CHOICES} \), and join each of those conjunctions with \{\text{FIRST}\} and \( \text{REMAIN_UNIQUE} \) to create all conjunctions of disjoint elements of \( \text{CHOICES} \), whose first elements is \( \text{FIRST} \) and cannot be extended with yet another disjoint element of \( \text{CHOICES} \).

The conjunctions generator can incorporate the covering test, so that the generator can be stopped if an exact covering of \( \text{CANDIDATES} \) is found.
The covering test of stage 3 can be implemented rather efficiently in the following manner:

- Each instance of the set \( CANDIDATES \) is transformed to a set \( COMPACT\_CANDIDATES \) by recursively coalescing blocks (remember that each element describes a block in a multi-dimensional space!), in which the order is that of decreasing numbers of included network nodes.

- Test whether \( COMPACT\_CANDIDATES \) covers \( DESTINATION \) entirely by using a recursion algorithm: Let \( HEAD \) be the first element of \( COMPACT\_CANDIDATES \) and \( TAIL \) the set of all other elements. Split \( DESTINATION \setminus HEAD \) into blocks and test whether \( TAIL \) covers all these blocks. Generally, combinatorial explosion will not arise, because we coalesced as many blocks specified in \( CANDIDATES \) as possible and ordered the set \( COMPACT\_CANDIDATES \).

We emphasized the combinatorial problems emerging in the analysis stage. The benefits of the methods presented in this section will become particularly useful if the number of characteristics occurring in the specification of the nodes of the network would be increased.

4 Final Remarks

We presented a general search method that can be implemented with little risk of combinatorial explosion. Thus the method is suited to be used interactively. The method supposes a state-oriented planning situation based on a network structure. Some more details of the approach are given for its application in tactical and strategic manpower policy making.

The main goal, however, of this paper was to demonstrate that for some situations it is sensible to combine rule-based with algorithmic approaches and let both approaches be responsible for activities in which they are strongest. In the type of problems described in the present paper, the algorithmic approach is clearly strongest for evaluating given scenarios. However, the rule-based approach is strongest in defining sensible scenarios, because of the complex relations between different aspects of scenarios and their consequences.
References


<table>
<thead>
<tr>
<th>Number</th>
<th>Month</th>
<th>Author</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>92-01</td>
<td>January</td>
<td>F.W. Steutel</td>
<td>On the addition of log-convex functions and sequences</td>
</tr>
<tr>
<td>92-02</td>
<td>January</td>
<td>P. v.d. Laan</td>
<td>Selection constants for Uniform populations</td>
</tr>
<tr>
<td>92-04</td>
<td>February</td>
<td>H.J.C. Huijbents, H. Nijmeijer</td>
<td>Strong dynamic input-output decoupling: from linearity to nonlinearity</td>
</tr>
<tr>
<td>92-05</td>
<td>March</td>
<td>S.J.L. v. Eijndhoven, J.M. Soethoudt</td>
<td>Introduction to a behavioral approach of continuous-time systems</td>
</tr>
<tr>
<td>92-06</td>
<td>April</td>
<td>P.J. Zwietering, E.H.L. Aarts, J. Wessels</td>
<td>The minimal number of layers of a perceptron that sorts</td>
</tr>
<tr>
<td>92-07</td>
<td>April</td>
<td>F.P.A. Coolen</td>
<td>Maximum Imprecision Related to Intervals of Measures and Bayesian Inference with Conjugate Imprecise Prior Densities</td>
</tr>
<tr>
<td>92-08</td>
<td>May</td>
<td>I.J.B.F. Adan, J. Wessels, W.H.M. Zijm</td>
<td>A Note on “The effect of varying routing probability in two parallel queues with dynamic routing under a threshold-type scheduling”</td>
</tr>
<tr>
<td>92-10</td>
<td>May</td>
<td>P. v.d. Laan</td>
<td>Subset Selection: Robustness and Imprecise Selection</td>
</tr>
<tr>
<td>92-11</td>
<td>May</td>
<td>R.J.M. Vaessens, E.H.L. Aarts, J.K. Lenstra</td>
<td>A Local Search Template (Extended Abstract)</td>
</tr>
<tr>
<td>92-12</td>
<td>May</td>
<td>F.P.A. Coolen</td>
<td>Elicitation of Expert Knowledge and Assessment of Imprecise Prior Densities for Lifetime Distributions</td>
</tr>
<tr>
<td>92-13</td>
<td>May</td>
<td>M.A. Peters, A.A. Stoorvogel</td>
<td>Mixed $H_2/H_\infty$ Control in a Stochastic Framework</td>
</tr>
<tr>
<td>Number</td>
<td>Month</td>
<td>Author</td>
<td>Title</td>
</tr>
<tr>
<td>--------</td>
<td>-------</td>
<td>--------------------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>92-14</td>
<td>June</td>
<td>P.J. Zwietering</td>
<td>The construction of minimal multi-layered perceptrons: a case study for sorting</td>
</tr>
<tr>
<td></td>
<td></td>
<td>E.H.L. Aarts</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>J. Wessels</td>
<td></td>
</tr>
<tr>
<td>92-15</td>
<td>June</td>
<td>P. van der Laan</td>
<td>Experiments: Design, Parametric and Nonparametric Analysis, and Selection</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>92-16</td>
<td>June</td>
<td>J.J.A.M. Brands</td>
<td>On the number of maxima in a discrete sample</td>
</tr>
<tr>
<td></td>
<td></td>
<td>F.W. Steutel</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>R.J.G. Wilms</td>
<td></td>
</tr>
<tr>
<td>92-17</td>
<td>June</td>
<td>S.J.L. v. Eijndhoven</td>
<td>Introduction to a behavioral approach of continuous-time systems part II</td>
</tr>
<tr>
<td></td>
<td></td>
<td>J.M. Soethoudt</td>
<td></td>
</tr>
<tr>
<td>92-18</td>
<td>June</td>
<td>J.A. Hoogeveen</td>
<td>New lower and upper bounds for scheduling around a small common due date</td>
</tr>
<tr>
<td></td>
<td></td>
<td>H. Oosterhout</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>S.L. van der Velde</td>
<td></td>
</tr>
<tr>
<td>92-19</td>
<td>June</td>
<td>F.P.A. Coolen</td>
<td>On Bernoulli Experiments with Imprecise Prior Probabilities</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>92-20</td>
<td>June</td>
<td>J.A. Hoogeveen</td>
<td>Minimizing Total Inventory Cost on a Single Machine in Just-in-Time Manufacturing</td>
</tr>
<tr>
<td></td>
<td></td>
<td>S.L. van de Velde</td>
<td></td>
</tr>
<tr>
<td>92-21</td>
<td>June</td>
<td>J.A. Hoogeveen</td>
<td>Polynomial-time algorithms for single-machine bicriteria scheduling</td>
</tr>
<tr>
<td></td>
<td></td>
<td>S.L. van de Velde</td>
<td></td>
</tr>
<tr>
<td>92-22</td>
<td>June</td>
<td>P. van der Laan</td>
<td>The best variety or an almost best one? A comparison of subset selection procedures</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>92-23</td>
<td>June</td>
<td>T.J.A. Storcken</td>
<td>Extensions of choice behaviour</td>
</tr>
<tr>
<td></td>
<td></td>
<td>P.H.M. Ruys</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>92-25</td>
<td>July</td>
<td>P.J. Zwietering</td>
<td>Exact Classification With Two-Layered Perceptrons</td>
</tr>
<tr>
<td></td>
<td></td>
<td>E.H.L. Aarts</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>J. Wessels</td>
<td></td>
</tr>
<tr>
<td>92-26</td>
<td>July</td>
<td>M.W.P. Savelsbergh</td>
<td>Preprocessing and Probing Techniques for Mixed Integer Programming Problems</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number</td>
<td>Month</td>
<td>Author</td>
<td>Title</td>
</tr>
<tr>
<td>---------</td>
<td>-----------</td>
<td>-------------------------------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>92-27</td>
<td>July</td>
<td>I.J.B.F. Adan, W.A. van de Waarsenburg, J. Wessels</td>
<td>Analysing $E_k</td>
</tr>
<tr>
<td>92-28</td>
<td>July</td>
<td>O.J. Boxma, G.J. van Houtum</td>
<td>The compensation approach applied to a $2 \times 2$ switch</td>
</tr>
<tr>
<td>92-29</td>
<td>July</td>
<td>E.H.L. Aarts, P.J.M. van Laarhoven, J.K. Lenstra, N.L.J. Ulder</td>
<td>Job Shop Scheduling by Local Search</td>
</tr>
<tr>
<td>92-30</td>
<td>August</td>
<td>G.A.P. Kindervater, M.W.P. Savelsbergh</td>
<td>Local Search in Physical Distribution Management</td>
</tr>
<tr>
<td>92-31</td>
<td>August</td>
<td>M. Makowski, M.W.P. Savelsbergh</td>
<td>MP-DIT Mathematical Program data Interchange Tool</td>
</tr>
<tr>
<td>92-32</td>
<td>August</td>
<td>J.A. Hoogeveen, S.L. van de Velde, B. Veltman</td>
<td>Complexity of scheduling multiprocessor tasks with prespecified processor allocations</td>
</tr>
<tr>
<td>92-33</td>
<td>August</td>
<td>O.J. Boxma, J.A.C. Resing</td>
<td>Tandem queues with deterministic service times</td>
</tr>
<tr>
<td>92-34</td>
<td>September</td>
<td>J.H.J. Einmahl</td>
<td>A Bahadur-Kiefer theorem beyond the largest observation</td>
</tr>
<tr>
<td>92-35</td>
<td>September</td>
<td>F.P.A. Coolen</td>
<td>On non-informativeness in a classical Bayesian inference problem</td>
</tr>
<tr>
<td>92-36</td>
<td>September</td>
<td>M.A. Peters</td>
<td>A Mixed $H_2/H_\infty$ Function for a Discrete Time System</td>
</tr>
<tr>
<td>92-37</td>
<td>September</td>
<td>I.J.B.F. Adan, J. Wessels</td>
<td>Product forms as a solution base for queueing systems</td>
</tr>
<tr>
<td>92-38</td>
<td>September</td>
<td>L.C.G.J.M. Habets</td>
<td>A Reachability Test for Systems over Polynomial Rings using Gröbner Bases</td>
</tr>
<tr>
<td>92-39</td>
<td>September</td>
<td>G.J. van Houtum, I.J.B.F. Adan, J. Wessels, W.H.M. Zijm</td>
<td>The compensation approach for three or more dimensional random walks</td>
</tr>
<tr>
<td>Number</td>
<td>Month</td>
<td>Author</td>
<td>Title</td>
</tr>
<tr>
<td>--------</td>
<td>-----------</td>
<td>-------------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>92-40</td>
<td>September</td>
<td>F.P.A. Coolen</td>
<td>Bounds for expected loss in Bayesian decision theory with imprecise prior probabilities</td>
</tr>
<tr>
<td>92-41</td>
<td>October</td>
<td>H.J.C. Huijberts, H. Nijmeijer, A.C. Ruiz</td>
<td>Nonlinear disturbance decoupling and linearization: a partial interpretation of integral feedback</td>
</tr>
<tr>
<td>92-42</td>
<td>October</td>
<td>A.A. Stoorvogel, A. Saberi, B.M. Chen</td>
<td>The discrete-time $H_\infty$ control problem with measurement feedback</td>
</tr>
<tr>
<td>92-43</td>
<td>October</td>
<td>P. van der Laan</td>
<td>Statistical Quality Management</td>
</tr>
<tr>
<td>92-44</td>
<td>November</td>
<td>M. Sol, M.W.P. Savelsbergh</td>
<td>The General Pickup and Delivery Problem</td>
</tr>
<tr>
<td>92-45</td>
<td>November</td>
<td>C.P.M. van Hoesel, A.P.M. Wagelmans, B. Moerman</td>
<td>Using geometric techniques to improve dynamic programming algorithms for the economic lot-sizing problems and extensions</td>
</tr>
<tr>
<td>92-46</td>
<td>November</td>
<td>C.P.M. van Hoesel, A.P.M. Wagelmans, L.A. Wolsey</td>
<td>Polyhedral characterization of the Economic Lot-sizing problem with Start-up costs</td>
</tr>
<tr>
<td>92-47</td>
<td>November</td>
<td>C.P.M. van Hoesel, A. Kolen</td>
<td>A linear description of the discrete lot-sizing and scheduling problem</td>
</tr>
<tr>
<td>92-48</td>
<td>November</td>
<td>L.C.G.J.M. Habets</td>
<td>A Reliable Stability Test for Exponential Polynomials</td>
</tr>
<tr>
<td>92-49</td>
<td>November</td>
<td>E.H.L. Aarts, J. Wessels, P.J. Zwietering</td>
<td>The Applicability of Neural Nets for Decision Support</td>
</tr>
<tr>
<td>92-50</td>
<td>December</td>
<td>F.P.A. Coolen, M.J. Newby</td>
<td>Bayesian Reliability Analysis with Imprecise Prior Probabilities</td>
</tr>
<tr>
<td>92-51</td>
<td>December</td>
<td>M.J.L. Hautus</td>
<td>Operator substitution</td>
</tr>
<tr>
<td>92-52</td>
<td>December</td>
<td>H. Boers, W.Z. Venema, J. Wessels</td>
<td>An Advisor Module for Tactical and Strategic State-Oriented Planning Systems with an Application in Manpower Planning Planning</td>
</tr>
</tbody>
</table>