USING EMPIRICAL KNOWLEDGE FOR THE DETERMINATION OF CLIMATIC SETPOINTS: AN ARTIFICIAL INTELLIGENCE APPROACH

Roger MARTIN-CLOUAIRE*
Thierry BOULARD**
Marie-Josée CROS*
Benoit JEANNEQUIN***

*INRA, Laboratoire d'Intelligence Artificielle, BP27, 31326 Castanet, France
** INRA, Station de Bioclimatologie, 84140 Montfavet, France
*** INRA, Domaine Horticole d'Alénya, Le Mas blanc, 66200 Elne, France

I. INTRODUCTION

Farmers of the next decade must become expert managers of all aspects of their farming operations. Good management will be more intensive and more demanding in terms of time and expertise, and more critical to farm survival. Automating or at least supporting the decision making process underlying agricultural production management is becoming an increasingly important issue in order to master the inherent complexity of the task and make a more rational use of resources. This applies in particular to greenhouse production systems because of the intricacy of the various processes involved and the manifold interactions between the phenomena pertaining to the biological characteristics of the crop and to the physical characteristics of the greenhouse. Moreover, the fast paces at which greenhouse technology, practical know-how and the range of plant varieties evolve do not let the grower acquire a robust management know-how by simple accumulation of experience.

The better management decisions that a knowledgeable and efficient computer system is capable to provide can be translated into higher yields, earlier produce and lower production costs. Rough estimations concerning the French Mediterranean context in which we are interested showed that for the same agronomic objectives (in terms of quantity, quality and timing) up to 25% of fuel energy could be saved by a more careful and wiser management of the greenhouse climate.

The work reported here concerns a computerized approach to support the tactical management of a greenhouse production system yielding tomatoes

which are grown during winter and early spring time in the southern part of France. A computer program, called SERRISTE\(^1\), has been developed for choosing automatically appropriate daily climatic setpoints to be maintained by specific devices equipping the greenhouse (heating and aeration systems).

Quantitative modeling of the whole crop and greenhouse systems is hitherto out of reach. For this reason the approach underlying the SERRISTE system rests on a rich body of empirical knowledge that would be used by a highly qualified grower in order to solve the setpoint determination problem. Simply stated, the setpoint determination problem consists in finding values that can ensure a satisfactory balance over conflicting goals and preferences such as: avoiding important damages to plants (e.g. diseases), maintaining good growth-development combination, minimizing heating-energy spending. The appropriate way to achieve these goals is predominantly dependent on the prevailing crop and weather conditions of the current day and on past and predicted conditions. Conventional daily management in greenhouses is based on predetermined sequences of setpoints (i.e. blueprint) that depend on static characteristics (e.g. latitude, equipments) of the greenhouse and the crop (e.g. variety of tomato). These setpoints vary with time, within the 24 hours of a day and along the cultivation period. Actually, they are just guidelines that must be adjusted by the growers depending on the crop conditions observed or predicted on a short term basis. Through such a tuning the growers try to perform a kind of empirical optimization of the dynamic behavior of the production system. The difficulty in doing this tuning stems from the strong interactions that exists between the effects of the actions associated with the setpoints. Moreover, the crop systems are poorly understood compared to many physical systems and, as usual with agronomical phenomena, much of our understanding is qualitative. Consequently taking the truly optimal decision in the setpoint determination problem is practically illusive.

Since quantitative approaches (e.g. optimization algorithms, simulation models) are not yet applicable, we have tackled the problem within an artificial intelligence framework that allows an explicit representation of the pertinent pieces of knowledge and a clear separation between the domain-specific know-how and its use in an intelligible reasoning process.

This paper describes the overall approach of the SERRISTE system with particular emphasis on the presentation of the body of knowledge incorporated in the system and the artificial intelligence techniques underlying the resolution process of the setpoint determination problem. The main part of the empirical know-how taken into account in the setpoint determination process is expressed through numerical constraints (relations) on the variables involved in the problem. SERRISTE solves what is called a constraint satisfaction problem (CSP) which involves the assignment of values to the variables subject to the above-mentioned constraints. Each set of assignments is considered as acceptable and is then evaluated and ranked according to preference criteria (e.g. economics). SERRISTE belongs to the class of knowledge-based systems and is essentially implemented in an object-oriented programming spirit. It runs on conventional PC-computers under the MS-DOS operating system.

The greenhouse decision and the computer frameworks in which the present work is inserted are introduced in Section II. Section III describes the fundamental elements of the body of empirical knowledge involved in the

\(^1\) SERRISTE means greenhouse grower in French.
setpoint determination problem. The artificial intelligence approach of the SERRISTE system is exposed in Section IV. Preliminary evaluations of the current version and further developments are are briefly discussed in Section V.

II. GREENHOUSE DECISION AND COMPUTING FRAMEWORKS

In agreement with the widely spread decomposition (Udink ten Cate and Challa, 1984; Baille et al., 1990) of the decisions that lie between strategy formulation and its implementation, we consider a decision process structured in a tree level hierarchy going at the bottom (level 1) from on-line climate control through (level 2) the tactical decision level concerning the determination of daily setpoints and to the higher (level 3) seasonal planning. Each level is associated with a particular goal and some decisions must be taken so that these objectives are satisfied. The decision making process at a particular level determines the objectives assigned to the next lower level except at the bottom one where decisions are directly transformed into actions (Cros and Martin-Clouaire, 1990). The first level encompasses a set of regulation algorithms that aims at controlling the important (and controllable) climate factors. The automation of the short term decision process at this level is fairly well mastered (although some improvements are still possible) in modern, well-equipped greenhouses. The second level deals with deciding the appropriate daily settings depending on the actual status of the crop (growth stage, vigor), weather conditions and timing situation with respect to the overall planning specified at this level. The decisions involved concern the environment setpoints of a particular day but the reasoning needed to reach these decisions must span a several day time scale. Only little research (Challa et al., 1989; Tantau, 1989) has been devoted to the issue of automating middle level decision making. Finally, given the general goal of maximizing the profit of the grower, the upper level, that is still to be explored, deals with the appropriate decomposition of the crop production season in growth stages and the determination of the corresponding mean inside temperatures according to constraints related to mean outside climate, crop growth and development, equipment, cost of production, tomatoes pricing conditions, etc...

Again, hitherto, only the first level is fully automated. In practice, this is realized thanks to the now classical distributed processing architecture in which real-time constraints on particular greenhouse compartments are handled by dedicated control computers that are connected to a central machine. The latter is typically of the PC type and is essentially used as an information system for collecting, storing and displaying environmental data. Beside its utilization for monitoring the important parameters of the different compartments it can be used as the media to modify the settings of the control devices that are regulated by the dedicated control computers.

The higher the level in the hierarchy the less often the decisions must be taken and the less structured the decision types. Thus, it is not surprising that the only really automated level is the first one. The lack of structure of a decision type means that either all steps of the decision cannot be specified before the decision is taken, or the decision depends on qualitative (i.e. incomplete)
knowledge or imprecise/uncertain data (e.g. predicted weather). Consequently, such decision types do not yield easily to formal analytical treatment by quantitative decision models.

The work reported here is an attempt to address the middle level decision process (daily setpoint determination) outside of the realm of purely quantitative approaches (e.g. optimization algorithms, simulation models) by relying on a rich core of heuristic knowledge. The knowledge-based system SERRISTE developed for this purpose has been designed to run on the central PC machine of the greenhouse computer environment.

III. BASIC KNOWLEDGE INVOLVED IN THE DETERMINATION OF CLIMATIC SETPOINTS

This section presents an analysis of the basic body of practical knowledge to take into account and how to use it wisely in the determination of daily climatic setpoints. The empirical knowledge we want to articulate pertains to modern greenhouses located in southern France and to winter crops (typically, tomatoes are planted in November and are grown through June with harvest beginning as early as mid-January). Such information is hard to find (very disseminated and incomplete) in the specialized literature because much of it is context dependent (i.e. specific to the climate in the area) or is still informal due to uncertainties pervading many aspects of the subject. This paper focuses only the climate issue. Other important components of the production management problem such as carbon dioxide concentration and fertilization are not considered here, although we agree that preferably they should not be treated independently of the others.

Subsection A presents the important aspects that have to be dealt with in the determination of climatic setpoints. Subsection B provides time decompositions of the days and growing season that are suitable for the task at hand. The order in which the decision process considers the different aspects involved is outlined in Subsection C. Finally Subsection D shows that the pieces of knowledge may readily be translated into constraints and rules.

A. Main factors to care of

In the problem of climatic setpoint determination one has to take care of and integrate the following two classes of variables describing:
- the situation outside the greenhouse, expressed through air temperatures, solar radiation, wind speed;
- the inside situation that we want to manage and which may be evaluated through both quantitative measures of air temperature, saturation deficit and soil temperature, and a qualitative appraisal of some physiological aspects of plants including especially the stage indicators and symptoms of diseases, wilting or too strong vigor.

The setpoints are another class of variables whose values specify the domains outside of which the control computers must command the use of devices such as the heating system (start or stop) or the roof windows (close or
open). The setpoint class includes the air heating, soil heating and aeration setpoints. Some elements of the above second class are directly associated with setpoints (e.g. air temperature and the air heating setpoint or air temperature and the aeration setpoint) and some are not (e.g. saturation deficit, vigor).

The choice of setpoints must be done by a reasoning process integrating the above-mentioned variables in an advantageous manner ensuring a profitable, though safe, combination of growth and development factors while keeping the energy spending within acceptable bounds and as low as possible.

Managing the production aims first at controlling the basic physiological functions such as photosynthesis, respiration, assimilation and transpiration that underlie the growth and development of plants. How the above functions individually depend on climatic conditions is rather well known though not sufficiently precisely for a accurate global quantitative modeling (Jones et al., 1989). Roughly, one can consider that growth is essentially affected by the intensity and duration of solar radiation that provides the energy needed in the photosynthesis process, whereas development is directly linked to the amount of heat (temperature) received over a period. Temperature influences also the rate of photosynthesis and, thus, the rate of growth.

The main elements characterizing the inside climate (light, temperature, saturation deficit) interact with each other; few interventions (heating, ventilating, shading) act preferentially on one particular variable but modify also several others. For examples, heating affects temperature but also the saturation deficit; ventilating affects both temperature and the saturation deficit and modify the carbon dioxide concentration (though we shall not consider explicitly this variable in this paper). Each decision has complex repercussions that are sometime opposed in their more or less delayed effects on the crop.

In order to reach the most advantageous decision in the setpoint determination problem one must rely on the currently available understanding of the phenomena, their interactions and above all the adequate decision attitudes in front of the possible classes of problems. Actually the main rules that must be fulfilled by the values of the variables in order to be part of an acceptable (i.e. rational according to the decision maker) solution are known, though imprecisely, by expert growers. For instance, the results of experimental works give us guidance on the most beneficial and forbidden light/temperature combinations in a particular period of the year for suitable growth and development of plants. Solving the determination problem is not easy, however, because several decision alternatives may have to be explored before a solution is reached. The generation and evaluation of these alternatives may be the source of a significant combinatorial complexity that growers may have a hard time to cope with, contrary to computers.

Besides growth and development factors care must be taken beforehand to prevent undesirable situations. This concerns in particular the incidence and development of diseases (mainly grey mold caused by Botrytis cinerea) or infestations by parasites. Essentially the preventive management decisions must ensure that conditions of high humidity (low saturation deficit to be more exact) are avoided. Again the rules governing such decisions are empirically available.

Another kind of undesirable situation is the lost of balance between reproductive and generative functions, characterized by particular aspects (size, shape, color) of different organs (leaves, stems, flowers, inflorescence) developing in the upper part of plants. Note that the interpretations of these
aspects may be different depending on the development stage, climatic data of
the close past and the time of their observation. The common symptom that the
growers are able to perceive is a too strong vigor where an acceleration of
vegetative growth is occurring while at the same time the flower production is
decreasing. This vigor symptom may have been caused by fertilization or
climate-related problems but can be controlled by proper climate settings. Simple
rules, induced from the observed practices of experienced growers, tell that the
appropriate reaction in case of a too strong vigor is to increase the mean (over
the day) air temperature, decrease the soil temperature and lower the humidity
(i.e. increase the saturation deficit). A too weak vigor is also undesirable and can
be corrected by the converse actions.

A specificity of the weather in the French Mediterranean area of France is
that periods of beautiful days with intense solar radiation may alternate abruptly
with dark and windy days. A sharp and sudden change of weather is harmful to
the plants. It stresses them due to the inertia of some physiological functions
such as water absorption. The stress phenomenon occurs when a long period of
cloudy and damp days (during which plants have gotten used to a low
transpiration activity) is followed by a period of dry sunny days. A wilting of the
plants ensues and is noticeably visible at the moment when the solar radiation
and saturation deficit are important. It may appear during several consecutive
days while the plants are not used to the new climatic conditions. From a
physiological point of view this wilting phenomenon corresponds to a demand
of water from the environment (strong solar radiation and high saturation deficit)
that exceeds the amount absorbed through the roots. The stomatal regulation
causes then the stomata-closure which in turn impedes the photosynthesis
process. A natural solution to this problem is to prepare the plants to the change
of weather that can be foreseen using the commonly available local area
forecasts. The preparation consists in increasing the soil temperature (to increase
the water absorption through the roots) and increasing the saturation deficit (to
increase the water demand of the environment) the last day of the cloudy weather
period. One could also reduce the detrimental effect of such a sudden change by
increasing the soil temperature and decreasing the saturation deficit the first day
of the sunny period.

Finally in deciding the control regime, especially the temperature regime,
to be maintained in the greenhouse the decision maker must take into account not
only the requirements of the plant but also the cost of providing heat. This
economical factor depends on timing, yield of the production, expected level of
market prices of tomatoes and price of energy. So far we have not considered
the problem in its full generality where the above considerations fluctuate from
one year to the other. Implicitly a standard situation is assumed and the solution
corresponding to minimal spending in energy is preferred.

As shown in this subsection, the local area weather predictions are worth
considering in the quest for good management performance.

B. Stages of growth and periods in a day

It is clear that the above factors are more or less relevant or important
depending on the stage of growth of the plants. To each stage correspond
specific rules used in the decision process. We are considering four main stages
that practically correspond to uniform morphological objectives and management
recommendations. Each stage is characterized by a sum of degree-days that the plants must receive within the corresponding interval of time. Table 1 describes the four stages in the case of the Capello tomato variety.

Tomato plants are subject to life-cycles that demand special considerations. Therefore, we have decomposed the 24 hours of a day in four periods:
- from sunrise to sundown during which the objectives concerning photosynthesis and transpiration functions must be satisfied;
- the first part of the night where respiration and assimilation partitionship is taking place as a continuation of the photosynthesis process of the previous diurnal period;
- the main part of dark night that requires heating, although the plant activity is lessen (this is where energy saving can potentially be made by adequate management);
- the dawn where the plants have to be prepared to the next daylight period and where humidity has to be taken care of.

The starting point and duration of the above periods are changing throughout the season.

A table of suitable ranges of the inside variables has to be given for each stage of growth in the interval of time that is of interest (i.e. from planting time until late in the fourth stage when control is no longer possible due to high temperature and intense solar radiation) and for each of the four periods considered in a day.

C. Decision process

The management of the greenhouse climate by the growers comes down to finding the best compromise between the aforementioned factors underlying the different stages of growth. The decision process must absolutely ensure a correct handling of aspects related to important or irreversible damages that may be caused to the crop and to the formation and development of fruits: avoiding the forbidden zones of the climatic variables, take care of risks of diseases and crop infestations. Then, outside the possibility of occurrence of dramatic situations, the reasoning process must consider choices of setpoints that can maintain a good growth/development combination, an adequate photosynthesis/respiration balance, a normal vigor of plants and a fair balance of the auxiliary fauna. The choices must also take into account a possible sharp change of weather. Finally among the possible alternatives the decision process must select the best or preferred one with respect to the criteria considered. Usually this amounts to find out the cheapest one in terms of cost of energy to be provided by heating.

The decision process starts from an overall objective expressed as a desired mean temperature over the 24 hours of the day that chiefly depends on the solar radiation forecasted. This objective can be slightly adjusted to take into account the recent history of the crop growth. For instance, the grower may want to make it up for missing degree-days if he is lagging behind in terms of the desired sum of temperature over a growth stage. Conversely, he may wish to slow down the crop development if he is ahead on his degree-days program.
<table>
<thead>
<tr>
<th>Stage</th>
<th>Initial State</th>
<th>Duration</th>
<th>Morphological Objectives and Recommendations</th>
<th>Climate Management Objectives and Recommendations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>planting</td>
<td>3 weeks</td>
<td>maintain vegetative balance, develop root system</td>
<td>maintain solar radiation, temperature balance, avoid low humidity in the first part of the stage, help fruit setting</td>
</tr>
<tr>
<td>2</td>
<td>3rd truss flowering</td>
<td>4 to 5 weeks before 1st harvest</td>
<td>maintain vegetative balance, help fruit growth</td>
<td>maintain solar radiation, air and soil temperature balance, be careful with high humidity at end of the stage</td>
</tr>
<tr>
<td>3</td>
<td>6th truss flowering (2 weeks before 1st harvest)</td>
<td>4 weeks</td>
<td>prevent infection of Botrytis, maintain vegetative balance</td>
<td>optimize the choice of mean air temperature with respect to available solar radiation, perform more systematic aeration and drying, prepare plants to abrupt changes of weather</td>
</tr>
<tr>
<td>4</td>
<td>harvest of 2nd truss</td>
<td>16 weeks or more</td>
<td>maintain vegetative balance</td>
<td>maintain vegetative balance, nutrition are correctly supplied, prepare plants to abrupt changes of weather</td>
</tr>
</tbody>
</table>
Furthermore, the adjustment of the global objective may be made for anticipatory reasons. The decision process goes through successive refinement of the overall objective by generating consistent sub-objectives associated with smaller intervals of time. At the end of this time-based iteration the setpoints which are nothing but sub-objectives concerning each of the four elementary periods composing a day are reached.

D. From empirical knowledge to constraints and rules

As explained in the sequel, SERRISTE operates on pieces of knowledge that are essentially represented under the form of constraints on the variables involved. For instance, the suitable growth/development combinations are expressed by mathematical functions that associates a solar radiation intensity (which is an imposed datum) with an interval of acceptable air temperatures. This association varies along the season and depends on static factors like the latitude of the greenhouse site or the physical characteristics of the greenhouse. The way of setting heating and aeration in order to obtain a desired mean air temperature and humidity during day time is another example of an empirical and approximate know-how that ultimately is expressed by a numerical constraint on the difference between aeration and heating setpoints.

The prevailing crop and weather conditions of the current day are inducing numerical constraints on the possible values of some of the variables to consider (including setpoints). The context dependency of these constraints is best conveyed by rules. For example, if the plants are too vigorous, then increase the mean daylight air temperature of about one or two degrees, decrease the soil temperature of about two degrees and increase the saturation deficit. Another example is given by the following rule: if damp weather is forecasted enforce a smaller difference between aeration and heating setpoints (that is the way of ensuring that aeration will occur frequently thus preventing high humidity that may contribute to disease incidence).

The precise form of the constraints and the way they are processed in the SERRISTE system for finding climatic setpoints is the subject of the next section.

IV. THE SERRISTE SYSTEM

From an artificial intelligence point of view, solving a problem requires knowledge representation capabilities and an inference machinery appropriate to the reasoning task at hand. We have translated the problem of the determination of setpoints into what is known in artificial intelligence as a constraint satisfaction problem (CSP). In this framework, knowledge is expressed through algebraic constraints restricting the possible values of numerical variables. The core of the reasoning process consists in a resolution mechanism that searches all the values (of the variables) that are compatible with all the constraints.

This section describes the conceptual scheme of SERRISTE and the main
features of its implementation. Subsection A outlines the artificial intelligence approach underlying SERRISTE. Subsection B shows briefly how the knowledge is represented. A technical presentation of the main features of the resolution mechanism is given in Subsection C. The global functioning of SERRISTE is described in Subsection D. The implementation issue is briefly addressed in Subsection E.

A. Outline of the artificial intelligence basis of SERRISTE

The problem of determining climatic setpoints is seen as an allocation task in which a set of variables must be assigned values satisfying simultaneously a set of constraints that represent pieces of knowledge to be taken into account. The constraints specify directly or indirectly the allowed (or forbidden) values of the concerned variables.

The field of constraint satisfaction problems (CSP) has been intensively studied and used in artificial intelligence (AI) (Mackworth, 1987; Kanal and Kumar, 1988). A wide range of practical tasks including planning scheduling, design can be modeled in this framework. See (Stone et al., 1992) for an example of application in crop rotation planning.

The core resolution process consists in choosing a not yet instantiated variable and assigning it a value taken in its domain such that all constraints bearing on this variable are satisfied. In doing so, one may reach a dead end in which no value can be given to the variable under consideration. A backtracking is then necessary; one must withdraw the instantiation of a previously considered variable and assign a new value to it. When all variables have been instantiated: a solution is then found. However, there may be zero or several solutions. In the latter case, resolution may be carried on until all solutions are found. This basic principle of search in a set of alternatives is conceptually at the very heart of artificial intelligence. Many algorithms have been developed in AI for this heavily combinatorial kind of problem.

The essential goal of a CSP resolution mechanism is to make the best possible choices in the following two situations:
- selection of the next variable to be instantiated,
- determination of the value to be assigned to the selected variable.

In the SERRISTE system, these choices are made according to two guideline criteria: respecting the logic underlying the setpoint determination process so as to facilitate a step by step understanding of the resolution, and ensuring efficiency of the resolution.

The selection of variables relies on the use of a specific tree structure of variable clusters that is constructed beforehand and that tells anytime which variables to consider next given the last instantiated. This so-called evaluation tree is produced by a careful analysis of constraint-induced dependencies between variables. Each node of this tree is a cluster of strongly interdependent variables.

Usually, before achieving an instantiation, nothing tells how to identify the values that will lead to a global solution. At most one can use a filtering technique to reduce, for each variable, the number of alternatives to be explored. Such a technique aims at eliminating impossible alternatives (i.e. those leading to inconsistencies). The immediate effect of applying the filtering technique on the variables of a cluster is to reduce the domains (i.e. the set of candidate values) of
these, or some of these variables. Reducing the domains of a particular set of variables may result in that some of the previously possible values (i.e. that are assumed to be so) of other variables become impossible. The filtering technique propagates automatically this effect through all the concerned variables in the constraint network.

Finally, when several sets of possible values are found by the resolution mechanism (usually around ten, but up to one hundred in some cases), SERRISTE has to perform a sort of tradeoff reasoning by integrating preferences (e.g. attitude with respect to disease occurrence, energy saving strategy) that are not expressed by constraints of the above-mentioned type. This ultimate phase of problem solving that aims at returning the "best" solution among the acceptable ones is not detailed in this article. We shall focus mainly on the issue of determining all the acceptable solutions.

B. Knowledge representation

The first step in applying the constraint satisfaction approach to setpoint determination consists in identifying the variables involved, their associated domains and the relations (i.e. constraints) between these variables.

So far, all the variables involved in the SERRISTE system are numerical and most of them concern temperatures. For illustrative purposes, only temperature variables are used. The set of variables includes those associated with setpoints plus some additional ones, called intermediary variables, that are needed for the reasoning process and for constraint expression. All these variables have continuous domains that are nevertheless treated as if they were discrete for two practical reasons: the CSP approach makes sense only for discrete domains and a too high precision would be meaningless. The discretization reflects actually the level of granularity found in typical human reasoning that deals with the task at hand; two temperatures that are less than 0.5°C away of each other are not considered as significantly different.

It is assumed that the domain of each variable is known and given in the form of an interval which is interpreted by the resolution mechanism as a discrete and finite set of values. These domains depend on the variety of tomato and environmental context. For instance, the domain of ITP3 (average night temperature) for a Rondello variety in the phase preceding the flowering of the first bunch is the discrete interval [11, 18] such that two consecutive values differ by 0.5°C.

Formally, constraints are relations between variables. In SERRISTE they are defined intentionally by linear equations of the general form

$$\sum_{i=1,n} c_i X_i = F$$  \hspace{1cm} (1)

where the coefficients $c_i$'s are scalar values, the $X_i$'s are variables (each having a domain) and $F$ is an interval. In this equation the meaning of the '=' sign should be understood as: given a set of $n$ values $v_i$ assigned to the variables $X_i$, the constraint is satisfied if $\sum_{i=1,n} c_i v_i$ falls within the interval $F$.

Any constraint may be parametrized in order to take into account its variation through the cultivation period and its dependency on the specific conditions of the current day. Both the set $F$ and the coefficients may be linked to context dependent parameters.

Under this general form the following constraints can be represented:
- Unary predicates (e.g. in the early stage of development, the average diurnal temperature ITP1 must belong to [14, 28]),
- Linear equations (e.g. the average temperatures at dawn, night and dusk, denoted by ITP2, ITP3 and ITP4 respectively, must satisfy the relation 3. ITP2 + 8. ITP3 + 2. ITP4 - 13. ITN = 0 where ITN is the average temperature of nocturnal time),
- Inequalities (e.g. if the foreseen solar radiation is high, one must have 2 ≤ ITD - ITN ≤ 6 where ITD is the average temperature of diurnal time).

A convenient way to visualize the dependencies between the variables is to build a constraint network incorporating all the variables and constraints involved in the problem. Figure 1 is a simplified example of such a network.

An edge between two variables means that a binary constraint connects these variables. Several edges converging in one point represent an n-ary constraint (i.e. any constraint involving n variables) on the variables located at the extremities of the edges (i.e. circled items). Usually, unary constraints are omitted for preserving a good readability of the network.

C. Resolution process

As outlined in Subsection A, the problem of setpoint determination is an allocation task in which numerical values must be assigned to a set of variables that are connected to each other through constraints. For the efficiency and the intelligibility (i.e. understandability) of the resolution process, a preliminary processing consists in building a structure that will provide an appropriate clustering of the variables and an order for exploring the constructed clusters. The clustering techniques are presented in the subsection below. The overall resolution process combines a filtering technique (that aims at eliminating impossible values) with a search algorithm that keeps trying instantiations and withdrawing them (backtracking) when a failure is detected. The filtering and search algorithms are described in the last two subsections. For more details, see (Cros and Martin-Clouaire, 1991).
Clustering of variables

The idea underlying this preprocessing phase is to exploit the structural properties of the constraint network in order to get together the variables that are strongly interdependent. Intuitively, it is wiser to try a simultaneous instantiation of such a group of variables because it is an effective strategy that avoids many backtrackings, and it brings a focussing effect that eases the task of a user who wants to follow the resolution process. The clusters of variables are organized in a tree that can be explored in a systematic way, providing an appropriate order for the instantiation of the variables.

Let us first define some basic notions used in the following. Two constraints are connected if a particular variable is involved in both (e.g. C1(V1, V2) and C2(V2, V3, V4) are connected because they share the variable V2). Two variables are directly connected if they are involved in the same constraint. More generally, two variables X and Y are indirectly connected if there exists a chain of connected constraints such that the first one concerns X and the last one concerns Y (e.g. with C1(V1, V2), C2(V2, V3, V4) and C3(V3, V5), V1 is indirectly connected to V5).

Clusters of variables are achieved and structured into a tree by running an iterative procedure that examines exhaustively the constraint network starting from a given set of variables defined as the root cluster. This procedure, when applied to a cluster of variables, produces the clusters that immediately descend from it (i.e. its sons). Initially, a set of variables playing a particular role is defined deliberately as the root cluster. The above-mentioned procedure is applied to this root cluster and is then used repeatedly for each newly created cluster. Without going into details (see (Cros and Martin-Clouaire, 1991) for the full algorithm), let us describe the basic principles behind the construction of sibling clusters descending from a given cluster G.

Let C1, . . . , Cr be the constraints in which at least one variable of G is involved and that have not already been considered by the clustering procedure. Let us suppose that Cj is one of these constraints. δj(G, Cj) denotes the set of variables concerned by Cj but not contained in G. This is called the development of G according to Cj. The construction of new clusters is obtained by merging the developments associated with C1, . . . , Cr if one of the three criteria given below is satisfied, until merging between developments or newly merged developments is no longer possible. Two developments or merged developments δ1 and δ2 are merged if they satisfy one of the three criteria:

- δ1 and δ2 have a non-empty intersection;
- a variable of δ1 is directly connected to a variable of δ2 through a constraint that has not been considered yet (i.e. that has never been involved in the development of a group);
- a variable of δ1 is indirectly connected to a variable of δ2 through constraints that have not been considered yet.

In our simple example illustrated in Figure 1, the root cluster which is the starting point of the instantiation process includes only the variable IT1 that stands for the average temperature desired over a 24 hour period and has only one constraint involving IT1. The development of the root cluster according to this constraint is the set {ITD ITN}. Since no other development has to be taken into account, this set is the son cluster of the root. Two constraints involving ITD and ITN have to be considered. The developments of {ITD ITN} according
to these constraints are (ITP1) and (ITP2 ITP3 ITP4). As these two developments do not satisfy any of the merging criteria, they give birth to two clusters that in turn are considered in a similar fashion. In this simple example however, no merging actually takes place. Ultimately, this construction process provides a tree of clusters.

Let us assume that a particular systematic procedure for exhaustive exploration of a tree structure is given. Then, the tree of clusters may be used as it stands for the purpose of guiding the instantiation process. However, efficiency and user friendliness of the resolution process may be improved by arranging the order of the sons of each cluster. As far as efficiency is concerned, it is preferable to consider first the clusters containing variables that are difficult to instantiate. In addition, a particular arrangement may turn out to be closer to the grower's pattern of reasoning. These late arrangements are not automated because they are user-dependent.

The network in Figure 1 enables the construction of the tree structure shown in Figure 2. The numbers near the groups of variables indicate the order in which the clusters are explored. Technically, this order corresponds to a left to right depth-first search. In other words, given an instantiation of some intermediary variables, this strategy is intended to find out as soon as possible if there exists a possible instantiation for the setpoint variables which are contained in the terminal or leaf clusters.

Figure 2. Tree structure of arranged clusters

Filtering
This process aims at reducing the set of candidate values. In SERRISTE, a so-called Waltz (1975) filtering is used. A general study of this general technique which has been employed in several artificial intelligence applications was made by Davis (1987). Given a set of variables subjected to some constraints, the filtering detects and eliminates the values that cannot take part in a solution. In some cases it may discover that the problem does not have any solution. It relies on an intelligent propagation process that is iterated through the
constraints and the variables involved in these constraints.

Let $V_1, ..., V_n$ be the variables involved in constraint $C$ and $A_i$ the possible values for variable $V_i$. The Waltz filtering is based on an essential operation, embedded in the procedure \textit{Reduce_domain} which when applied to the constraint $C(V_1, ..., V_i, ..., V_n)$ and the variable $V_i$, returns the subset of $A_i$ obtained by removing all values incompatible with all conceivable sets of values that might be assigned to the other variables. A formal definition of this operation is:

\[
\text{Reduce\_domain}(C,V_i) = \{ a_i \in A_i \mid \exists a_j \in A_j, \ j=1,n, \ j \neq i \ C(a_1, ..., a_i, ..., a_n) \}
\]

where $C(a_1, ..., a_i, ..., a_n)$ means that the constraint $C$ is satisfied by the value $a_j$ for the variable $V_j$, $j = 1,n$.

For continuous variables, the programming of this procedure depends on the nature of the constraint concerned. If $C$ allows to express each variable as a function of the other variables (this is the case for the type of constraints manipulated by SERRISTE) and if each $A_i$ is an interval, then the \textit{Reduce\_domain} operation can be performed by a calculus on intervals which requires only simple computation on the bounds. For example, given the constraint $V_1 + V_2 = 10$, the initial domains $A_1 = [0, 2]$ and $A_2 = [5, 9]$. The new filtered domains $A_1'$, $A_2'$ can be obtained as follows: $A_1' = A_1 \cap [10 - 9, 10 - 5] = [1, 2]$ and $A_2' = A_2 \cap [10 - 2, 10 - 0] = [8, 9]$. Note that the result would not be improved by reusing \textit{Reduce\_domain} with $A_1'$ and $A_2'$ instead of $A_1$ and $A_2$ (i.e. the returned sets would not be smaller).

Such a filtering takes advantage of the linearity of the constraints and amounts to treat the variables as continuous. We shall see that the general search algorithm presented in the next subsection works with discrete variables. The domain discretization that is necessary to carry on the resolution is done after any filtering operation. It is a particularity of our approach to use the algebraic nature of constraints and variables to switch conveniently from continuous to discrete domains and vice versa. So, rather than an expensive filtering on discrete universes, SERRISTE performs a much simpler filtering using interval calculus on continuous universes.

In the Waltz algorithm, the \textit{Reduce\_domain} operation is used in the procedure \textit{Revise} which, for a given constraint $C$, detects among its associated variables those with a domain that can be reduced and eliminates the corresponding values. The Waltz algorithm applies to a set of constraints that one is willing to take into account. Each of these constraints are considered in turn by the \textit{Revise} procedure. Each time some values can logically be removed from the domain of a particular variable involved in the constraint under consideration, the algorithm removes these values. The other constraints involving variables with a modified domain are then pushed in the queue of constraints to be considered by the Waltz filtering. In this way, the consequences of reducing a domain are propagated and induce modifications in the domains (i.e. of other variables) through the appropriate constraints. The propagation of domain reduction is carried on until there is no more constraint to be considered (natural quiescence) or until a termination condition applies.

Termination criteria (such as the number of times any constraint may be considered) are needed because, in some cases, the Waltz filtering algorithm may go into infinite loops or reach quiescence only after a long time. Even in these
cases however, filtering is worth doing since it reduces the number of failures in the instantiation process. The Waltz algorithm is sound, that is, it cannot eliminate good values (it cannot cause a solution to be missed), but incomplete (except with simple constraints) since the reduced domains may still contain elements whose combination is incompatible with some constraints.

Search algorithm

The complete resolution mechanism first performs a filtering over the entire set of constraints in order to attain a certain level of consistency by pruning the values that for sure cannot take part in a global solution. At this stage a global inconsistency may be discovered. It means that in such a case there is no solution and there is no need to push the resolution further. If successful in the initial filtering the resolution mechanism enters in a typical CSP search algorithm that considers one cluster after the other (in the order specified by the tree structure built for this purpose). Each time, it makes tentative instantiations of the variables of the current cluster, checks that they satisfy the directly concerned constraints, propagates by Waltz filtering their effects on other variable domains and backtracks to another tentative or another group when a complete solution is found (so as to seek another one) or when a failure is encountered. This essential algorithm corresponds to the following procedure, called FindSolutions, which (hopefully) is self explanatory.

Procedure: FindSolutions

set CurrentCluster to root-cluster
1 if none of the domains of the variables of CurrentCluster is empty
   then chose a combination of values from each of these domains
      append it to the partial solution under development
      suppress these values in their corresponding domains
   else goto 4 (i.e. backtrack)
2 if any constraint concerning the current and/or already considered clusters
   is not satisfied
   then goto 1 (i.e. try another combination of values)
3 if filtering applied to the other concerned constraints uncovers inconsistency
   then undo this local filtering
      maintain (clean) the data structure of the partial solution
      goto 1 (i.e. try another combination of values)
4 if the CurrentCluster is the last one
   then save the partial solution under development
      (since it is indeed an acceptable solution)
      goto 1 (i.e. try another combination of values)
   else set CurrentCluster to the next one
      (according to the order induced from the tree)
      goto 1 (i.e. try another combination of values)
4 if the CurrentCluster is the root-cluster
   then return (i.e. all the acceptable solutions, if any, have been found then)
   else undo the local filtering
      (put back the domains as they were after the initial filtering)
      set CurrentCluster to the preceding one

\* Incidentally, this indicates that there is something wrong in the knowledge base since the system should be able to provide setpoints in any situation.
maintain (clean) the data structure of the partial solution

goto 1 (i.e. try another combination of values)

From an artificial intelligence point of view, the salient features of our approach are the processing of variables by clusters rather than individually and the incorporation of a filtering operation before any instantiation of the variables contained in a cluster. We also use advantageously the fact that the domains of the variables may be considered as continuous for filtering purposes and discrete in the final stage of search.

D. The global functioning of SERRISTE

The global functioning of SERRISTE can be summarized in a seven-step process. The first two steps (Phase A) are performed only once in a cultivation period. They consist in an initialization of some computational structures and collection of data that remain valid over the entire cultivation period. After these two steps have been completed the resulting structures and collected data are saved in an appropriate manner so as to be reused every day of the cultivation period. The sequence going from Step 3 to Step 7 (Phase B) corresponds to a typical routine (daily) use of SERRISTE for the determination of setpoints.

Step 1: "Compilation" of the knowledge base

This step aims at analyzing the knowledge base in order to set up some data structures mainly for efficiency of the resolution. In particular, this is where the variables are clustered into groups which are themselves organized in the so-called evaluation tree.

Step 2: Input of data pertaining to the greenhouse and the crop

The user has to answer to a sequence of questions concerning in particular:
- the location of the greenhouse (latitude);
- the characteristics of the greenhouse (size, type of heating system, type of cover, availability of thermal screens,...);
- variety of tomato used.

Step 3: Input of data pertaining to the current conditions

The questions concern:
- the current date and current stage of crop growth;
- a qualitative assessment of the state of the crop (vigor, presence of disease);
- the forecasted weather for the current day (upper and lower temperatures, wind speed, cloud cover in qualitative terms like cloudy, blue sky, rainy);
- the forecasted cloud cover for tomorrow;
- effective cloud cover observed yesterday;
- effective climate conditions observed (measured) yesterday inside the greenhouse;
- short term objectives (e.g. increase vigor, compensate a temperature backlog).

Step 4: Computation of needed information based on data obtained in Steps 2,3

From the data collected in the preceding steps the values of the parameters involved in the constraints and other useful parameters such as the time of sunrise, the estimated numerical value of maximal solar radiation are computed by applying appropriate formulae.
The domains of some variables vary depending on the stage of growth and the variety of tomato. The domains corresponding to the considered situation are deduced by firing appropriate rules.

**Step 5: Resolution of the constraint satisfaction problem**

In this step, which has been explained in Subsection C, SERRISTE makes an initial filtering to check the global consistency of the problem and ease the search of solutions. If the filtering does not discover an inconsistency the system enters the FindSolutions procedure. By the end of this step, the system has produced a (possibly empty) set of assignments to the variables.

**Step 6: Choice of the "optimal" assignment**

The system computes among all assignments (which represent acceptable solutions) the "best" one according to the specified criteria (typically minimize consumption of energy of the heating system). A very simple model of energy balance is used.

**Step 7: Save results and prepare reports according to the demand of the user**

The user may wish to keep a daily report file containing the chosen solution together with the conditions (input and computed) characterizing the current day. For analysis purpose, the user may request to save also additional information such as the set of all acceptable solutions or a detailed trace of the run which are very useful to pinpoint various kinds of problems (e.g. why there is no solution or which constraint does not work properly and is responsible for aberrant solutions).

### E. Basic principles of the implementation

The SERRISTE system has been developed with KAPPA (IntelliCorp, 1991) which is essentially a knowledge representation tool kit. It is a strongly object-oriented hybrid environment that allows to mix different features—objects, rules, functions, demons and conventional programming techniques—when developing an application. KAPPA’s objects can be structured in hierarchies composed of classes, subclasses and instances. The properties of an object are expressed in slots or attributes that may themselves be characterized by restrictions concerning for instance the type of information allowed as value (text, numerical, Boolean, object) or the number of values allowed in case of multivalued slot. The behavior of the objects is defined by procedural code in the so-called methods which can be inherited as any other property through hierarchical links. In addition, to any slot one can attach demons that automatically respond to alterations on the slot. The rules can be used in forward or backward modes to perform reasoning tasks involving the objects. The programming language KAL used within KAPPA (for writing methods for instance) is closely integrated with the C programming language, in that external C functions can be called directly. KAPPA (version 1.2) runs under MS-WINDOWS and the MS-DOS operating system.

The CSP-related capabilities of SERRISTE are embedded in four classes, *Quantities*, *Constraints*, *Solutions* and *Groupes*. *Quantities* has two subclasses: one for the variables involved in the constraints and one for the various kinds of parameters. The hierarchy below *Constraints* provides the prototypes of the different kinds of constraints. They are distinguished by the number of variables that they concern. The object *Solutions* has a subclass *SolutionsAcceptables*
whose purpose (as its name indicates) is to gather the different acceptable solutions found by the system. The slots of this object are numerical and have names corresponding to the variables. When a solution is found during the search process, an instance of SolutionsAcceptables, named SolutionAcceptable_n (n being the number of the solution) is created, and each slot is assigned the value of the corresponding variable. The subclass SolutionsPreferees is designed to receive the instance of the preferred solution (so far we have assumed there is only one optimal solution). Groupes is the prototype of a cluster of variables, i.e. its instances are the clusters involved in a particular application and are created in Step 1. There is also an important object, named Resolution, which involves the top level methods of the resolution process. The main properties and methods attached to these objects and the role they play in the resolution process are described in (Martin-Clouaire et al., 1992).

The aspects of the real world which are relevant and important for the climate management task are represented through objects such as Serre (meaning greenhouse), Culture (meaning crop) and Temps (meaning time). In addition, instances of the two classes Meteos (meaning weather) and ClimatsMesures (meaning measured climate data) express weather forecasts and greenhouse climate measurements for a given day. The instances are named Meteo_j or Climat_j, where j refers to the date of the considered day.

The essential part of knowledge specific to a particular application is expressed by instances of the classes Variables and Contraintes. This section gives concrete examples of two instances used in the current version of the knowledge base. For clarity, they are presented in a slightly edited manner to avoid the burden of syntax definition.

CTsol Instance of subclass Variables

Slots with their local values if any:

Valeur
TMinMin = 10
TMaxMax = 30
ValeurMin
ValeurMax
ValeurMinApresFiltrageInitial
ValeurMaxApresFiltrageInitial
ValeurMinReduite
ValeurMaxReduite

Local methods:

Figure 3. The slots of the instance CTsol

The object CTsol (Figure 3) represents the soil temperature setpoint variable. It is shown in a state where only two slots have received values. There is a slot Valeur that contains the value of the variable when available. The other slots go in pairs and convey lower and upper bounds of the variable value at four different stages of the resolution process. They correspond, in the order of Figure 3, to the default situation before any processing (e.g. a soil temperature
setpoint is always within the bounds 10 and 30), the post-initialization state, the situation once the initial filtering has been performed, the current situation while searching the acceptable solutions. Note that the instance \textit{CTsol} does not have any local method.

\begin{verbatim}
C10 Instance of subclass ContrainesBinaIres

Slots with their local values if any:
V1 = ITN
V2 = ITD
a1 = 1
a2 = -1
a3Min = 2
a3Max : Demon Determiner_a3Max

Local methods:

Determine a3Max
  { GetValue(DTD_N; Valeur); }
\end{verbatim}

Figure 4. The slots and method of the instance \textit{C10}

The constraint \textit{C10} represents the following piece of information: the difference between the average temperature of diurnal time (denoted ITD) and the average temperature of nocturnal time (denoted ITN) must be within the interval [2, ATD_N] (i.e. \(2 \leq ITD - ITN \leq ATD_N\)). The parameter \(ATD_N\) depends on the prevailing weather conditions of the day. The instance \textit{C10} (see Figure 4) is a member of the class \textit{ContrainesBinaIres} which has slots for the terms composing a typical binary constraint of the form \(a_1V_1 + a_2V_2 = [a_3\text{Min}, a_3\text{Max}]\). At the time of the definition of \textit{C10} the value of \(a_3\text{Max}\) is not known. So the demon method \textit{Determine_a3Max} has been associated to the corresponding slot and gets the value as soon as it is needed. The code of the local method \textit{Determine_a3Max} simply tells to get and return the value of the slot \textit{Valeur} of the object \textit{DTD_N} which represents a particular parameter.

\begin{verbatim}
Rule DCTsolAdapt6
If
  Culture:Botrytis = non And
  Culture:Vigueur = bonne And
  Culture:ObjVigueur = diminuer And
  Meteo_j-1.qMRgj = faible And
  Meteo_j-1.qPRgj = eleva
Then
  DCTsolAdapt.Valeur := 1
\end{verbatim}

Figure 5. The rule \textit{DCTsolAdapt6}

In the current knowledge base of SERRISTE, forward chaining rules are used to deduce the values of some parameters that depend on the data entered in
the initialization phases. The rule shown in Figure 5 tells literally that if the crop is not affected by Borylis and its vigor is good and the objective with respect to vigor is to make it lower and the solar radiation was poor yesterday and the solar radiation should be high today then, the slot Valeur of the parameter DCTisolAdapt must be set to 1.

As far as the size of the problem is concerned, the current knowledge base contains 24 variables, 43 parameters, 27 constraints and 62 rules.

V. DISCUSSION

The knowledge-based system SERRISTE aims at providing the versatile decision support capabilities that are needed for an efficient management of daily greenhouse climate. The main motivation behind the work reported here was to gather an adequate body of heuristic knowledge and develop the artificial intelligence software SERRISTE capable of representing this knowledge in a format appropriate to the execution of the task on conventional greenhouse computers. Before the advent of accurate mathematical models of crops and greenhouses becomes a reality it is our belief that the best practical source of help in the management task is provided by a mixture of a general scientific background (on plant physiology and greenhouse engineering) with empirical agronomic expertise about growing particular crops in particular greenhouses under specific conditions. See, however, the papers by (Jones et al., 1991) and (Seginier, 1991) for recent contributions on the side of quantitative approaches.

SERRISTE rests on a body of knowledge that is typical of what expert growers are using and that we have expressed essentially under the form of constraints on numerical variables. However, strictly speaking, SERRISTE is not mimicking human reasoning in solving the setpoint determination problem; the constraint satisfaction approach is too combinatorial to be carried out by a human brain. Thus, SERRISTE is more a knowledge-based system than an expert system.

Few works have been reported on computer decision support systems applied to the management of greenhouse climate. Of special interest, however, is the paper by (Gauthier and Guay, 1990) that presents an experimental object-oriented design of an ambitious system addressing both management and control of greenhouse climate.

A. Interesting features and current status

An expert-based decision making allows a rigorous management of interactive parameters such as humidity and temperature and, thus, greatly contributes to better prevent diseases (which otherwise cause expensive interventions). Moreover, the decision support capabilities of SERRISTE, if used regularly, bring along coherence in the crop management; harmful effects of discontinuity that are encountered when several decision makers are involved in the choices of setpoints can be avoided.

Besides relying on expert knowledge and AI constraint satisfaction techniques, the management approach undertaken through the SERRISTE
system is novel in that it exploits weather predictions for better satisfying the physiological requirements of the plants, and for reducing energy costs. It also takes into account qualitatively assessed data such as symptoms of diseases and vigor of the plants.

The structuring of expertise, that has been discussed in Section III, is a required step before incorporating the knowledge in a decision making computer system. The formalization of growers' know-how has brought valuable benefits. Experts make progress in the field because they have to think about their practices, clarify what is implicit and uncover gaps in knowledge. Another significant by-product of this basic work is that it contributes to facilitate the transfer of knowledge.

Basically, SERRISTE solves an assignment problem through a general artificial intelligence approach. Although so far the constraints have chiefly been linear, we did not use mathematical programming techniques for the following reasons. Firstly, a clear objective function is lacking (reaching the truly optimal solution is illusory given the qualitative nature of the knowledge to take into account). Secondly, it is highly desirable to produce an understandable resolution for users so as to facilitate the maintenance and development of the knowledge base. Finally, we wished to keep open the possibility of incorporating easily new pieces of knowledge and using heuristic choices if necessary.

The computer program centered on the constraint satisfaction techniques has been tested extensively to uncover and fix programming errors. We have started the validation process of the whole SERRISTE system, that is, the application-specific knowledge possessed by the system and the overall computational approach. We are now in the process of verifying the consistency and the quality of the results provided by SERRISTE. Indeed a currently undergoing experiment consists in applying the setpoints of SERRISTE in one compartment of a greenhouse and compare the results with those obtained in a neighbor compartment managed by a human grower that applies conventional rules. The experiment is not finished yet and it is not the subject of this paper to discuss its results but the available empirical evidence has shown so far positive benefit from using SERRISTE.

Although SERRISTE is presently only a prototype system the overall approach seems sound. Deeper testing and evaluation are required: the system must be confronted to a significant number of cases and it must be confronted with a larger problem including the management of carbon dioxide and nutrition. On the problem solving side SERRISTE suffers from deficiencies related to the following aspects. First and most importantly, the system is too slow. Second the expressive power of the knowledge representation capabilities is too limited. It is impossible to code faithfully the expert knowledge because of the inherent imprecision pervading it. The crisp constraint framework is not satisfactory in this respect. Third, sometimes there are too many solutions among which some are not significantly different.

B. Future works

The overall project of computer-aided greenhouse management undertaken through the SERRISTE system is, of course, far from complete. Currently, we focus the effort on an extension (Martin-Clouaire, 1992) of
the representation capabilities in order to deal adequately with soft constraints. Basically, we use approximate reasoning techniques that have their roots in the theory of possibility (or fuzzy set theory). In addition, this extension allows to make distinctions among the constraints; some being more important than others.

Other directions that are in need of further exploration and development are discussed in the remaining part of the section.

The current knowledge base needs to be improved on several aspects. In particular, what remains to be addressed is the problem of managing the production in the second half of the spring season characterized by higher outside temperature. As already mentioned, the present version of the system deals only with climate issues. The next step to consider is the management of carbon dioxide enrichment and nutrition conjointly with the climate factors.

Before considering a routine use of an extended system of the SERRISTE type one must carry out an in-depth validation analysis by using it in real contexts (real greenhouses) under various and numerous conditions. The robustness of the approach could be evaluated by applying it to other geographic contexts. Moreover, other crops may be considered.

In order to be really useful, a greenhouse management system must provide solutions (if any) or advice in critical or exceptional situations that usually leave growers in helpless positions. Such situations occur, for instance, when a particular device used to modify the greenhouse climate has not worked properly (due to a failure for example). The kind of knowledge required to cope with such problematic cases has still to be identified. Whether or not the current reasoning capabilities of SERRISTE are sufficient remains to be shown.

The exploitation by a decision making system of the property that tomato plants act as efficient temperature integrators over a several day period enables anticipatory decisions which can potentially contribute in a significant manner to energy saving. The rationale behind these "reservoir" capabilities of plants and the basic principles behind its profitable exploitation are briefly addressed in (Bouard et al., 1991). So far SERRISTE leaves the burden of reservoir management to the grower who is given the possibility to adjust the overall objective of the day (see Subsection D in Section III). Providing a decision support that could use the reservoir capabilities is difficult for several reasons including: the incompleteness of knowledge on the topic, the necessity to reason over a several-day time scale and, of course, to decide under uncertain predictions.

Acknowledgements: This work has been partially supported by the AFME agency (Agence Française pour la Maîtrise de l'Energie). It has been achieved in the framework of an INRA project involving colleagues from Bioclimatology (M. Mermier, P. Reich, S. Simon) and Agrarian Systems and Development (J. Lagier) Departments. J. Jihad, K. Kováts and V. Terrazzoni have contributed to the implementation of SERRISTE.

REFERENCES


