[14] **Ray Reiter**: "A Theory of Diagnosis from First Principles," *Artificial Intelligence* **32**(1), 1987

[15] **G. Scemmama**: "Réalisation d'un système à base de connaissances pour la gestion du traffic urbain," *Recherche Transports Sécurité*, June 1989

[16] **P. Struss, O. Dressler**: "Physical Negation - Integrating Fault Models into the General Diagnostic Engine," *Proceedings of the 11th IJCAI, Detroit*, 1989

[2] **R. Davis, W. Hamscher**: "Model-based Reasoning: Troubleshooting," *Exploring Artificial Intelligence* Morgan-Kauffman, 1988

[3] **J. DeKleer, B. Williams**: "Reasoning about Multiple Faults," *Artificial Intelligence* 1987, **32**

[4] **J. DeKleer, B. Williams**: "Diagnosis with Behavioral Models," *Proceedings of the 11th IJCAI*, 1989

[5] **J. DeKleer**: "An Assumption-based TMS," *Artificial Intelligence* **28**, 1986

[6] **J. DeKleer**: "Extending the ATMS," *Artificial Intelligence* **28**, 1986

[7] **J. DeKleer**: "Problem Solving with the ATMS," *Artificial Intelligence* **28**, 1986

[8] **J. Dekleer, J.S. Brown**: "A Qualitative Physics based on Confluences," *Artificial Intelligence* **24**, 1984

[9] **J. DeKleer, K. Forbus, D. McAllester**: "Truth Maintenance Systems," *Tutorial SA5, 11th IJCAI, Detroit* 1989

[10] **C. Joubel, O. Raiman**: "How Time Changes Assumptions," *Proceedings of the 9th European Conference on Artificial Intelligence*, Stockholm, 1990

[11] **F. Toledo, S. Moreno, F. Rosich, G. Martin**: "Qualtiative Simulation in Urban Traffic Control : Implementation of Temporal Features," *Proceedings of IMACS Workshop on Qualitative Reasoning in Decision-Aid Systems*, Toulouse, 1991

[12] **Ph. Glayre, Ph. Bovy**: "Régulation lumineuse des carrefours," *Cahier TEA N.2, Département de Génie Civil, EPFL*, 1988

[13] **P.B. Hunt, D.I. Robertson, R.D. Bretherton, R. I. Winton**: "SCOOT - a traffic responsive method of coordinating signals," /em Department of the Environment, Department of Transport TRRL Laboratory Report 1014, Crowthorne (UK), 1981

problems in the street network could be detected immediately based on inconsistencies in the sensor readings.

6 Conclusion

In this paper, we have described a novel approach to the difficult problem of urban traffic control. Using first-principles reasoning, control strategies are synthesized which overcome the weaknesses of optimization-based approaches and do not require heuristic knowledge of experts.

The strength of the approach is that knowledge about global phenomena does not have to be summarized in local heuristic rules, but is automatically synthesized by the ATMS. Even though local models are highly simplified, the fact that global consistency is exploited makes them very powerful. In a practical application, the model-based approach could be combined with an optimization strategy based on more precise models to combine the advantages of both paradigms.

Another contribution of this paper is to extend the model-based diagnosis paradigm established by GDE ([3]) to model-based *control* which can be continuously applied. This possibility is worthy of further investigation in other domains where complex control strategies must be developed, such as process control or financial investments.

The current prototype is built with the goal to explore the concept of model-based control. Its performance is very promising for further development of the technique, since it has shown that the idea of ensuring global consistency of a control strategy using model-based reasoning is computationally feasible. In order to avoid the political and practical difficulties with applications in traffic control, we are planning to investigate applications of these techniques in other domains, such as the control of communication networks.

References

[1] **J. Cuena**: “Buildling Expert Systems based on Simulation Models: An Essay in Methodology,” in: L. Bole, M.J.Coombs (eds.): *Expert System Applications*, Springer Verlag, 1988

extending the choices of our system beyond the fixed timeplans constructed by traffic engineers should allow substantially better performance of our system. Another possibility for significant improvements is to reason explicitly about synchronization between intersections.

The design of timeplans is a complex constraint satisfaction problem which is difficult to solve by hand. Even with current control systems, the model-based reasoning technique might be useful to adapt timeplans to changes in the traffic network caused by weather, roadwork or accidents.

The main reason why our first prototype is based on fixed timeplans is the availability of realistic test data. We hope that cooperation with traffic engineers will allow us to experiment with different models in the future.

5.2 Temporal Reasoning

The strategies generated by our current system are largely *reactive*: new control regimes are chosen only in response to problems. In reality, it is desirable to *avoid* problems before they occur. Our system can be adapted to handle this case by replacing the measurements with measurements predicted into the future. Note that because of the structure of the system, the predictions do not have to be precise: ambiguities simply lead to multiple consistent solutions. A promising way to obtain predictions is that of qualitative simulation ([11]).

An explicit model of temporal evolution would also allow more explicit formulation of control *strategies*. However, from our experience with the current system, it seems that the instantaneous (and implicit) strategies are already very powerful. Long-term strategies may not be much more successful because of the unpredictability of traffic.

5.3 Diagnosis of Traffic Incidents

An important problem in traffic monitoring is to detect incidents such as traffic accidents, or faults in traffic sensors such as cars parked on top of them. This *incident detection* is a major problem of traffic engineers today, as such problems often go undetected (and uncorrected) for long periods of time and cause havoc in the traffic network. By extending the diagnostic capabilities from control models to topological models, accidents or other lanes.

The fact that our strongly simplified models can already compete favorably with the solutions generated by human traffic engineers means that the approach is a promising one for full-scale implementation. It seems likely that with more flexible timeplans and explicit representations of synchronization in our models, significant improvements even over the currently optimal fixed strategy could be achieved.

As far as the behavior of the system is concerned, we note that the system consistently generated reasonable and well-constrained solutions (apart from the optimal case already discussed). It takes about 140 seconds of machine time to run twenty minutes of real time simulation, out of which about 93 % is spent in the simulator[1]. Considering that the center city of Lausanne as a whole contains 82 intersections (63 with traffic light control), we thus do not expect the computational complexity of the procedure to be a problem in practice.

A big difficulty with evaluating our methods is the availability of realistic traffic data, which is very costly to obtain. The example network shown in Figure 8 is the only network for which actual measurements were available, and then only for a 20 minutes of actual traffic. Furthermore, it is not clear how realistic traffic simulations actually are. The only real test could be full-scale implementation of such a control strategy in an actual traffic control system, which for reasons of cost is not available to us.

5 Future Work

While our first prototype already shows promising results, it does not yet exploit fully the advantages of our formalism. In this section, we discuss three important extensions which will make future prototypes much more attractive.

5.1 Flexible Control Models

In section 2.3, we have discussed the possibility of flexible control models which would allow much more powerful control. As we have seen in the example of the city of Lausanne,

---

[1] We use an event driven micro simulator, which simulates individual car movements and is therefore rather slow.

simul type	*Collapsed lanes*	*Average waiting time*	*cars waiting*
1	3.33	27.7 sec	430 cars
2	3	30.5 sec	385 cars
3	9	40.6 sec	816 cars
4	3.5	36.9 sec	398 cars

Table 4: Simulation of part of Lausanne, result obtained

4. simulation with the same (random) control settings as above, but with adaptation using our system.

The performance is reflected in the statistics summarized in Table 4. They characterize the traffic situation by the number of lanes with *collapsed* traffic flow at the end of the simulation, the average time a car spends waiting in queues, and the total number of cars waiting in any queue. Even compared with the optimized and well-synchronized control settings, our systems performs very well: 9 % less blocked lanes, 8.7 % more average waiting time and 11.7 % less cars waiting. Only the average waiting time is slightly higher than that obtained with the controls optimized by traffic engineers. This is a very promising result, as traffic engineers have difficulty to achieve even such slight improvements over their already optimized timeplans. It is suprising when we consider that our system is handicapped by the fact that it does not explicitly represent synchronization. We conjecture that the loss of synchronization and the corresponding reduction in street capacities is responsible for the higher average waiting time. On the other hand, the fact that the time wasted in the lanes is higher, but the number of cars waiting is lower tends to prove that the overall capacity of the network is used in a better way: queues are more evenly distributed over the network.

The third and fourth simulations compare the performance of our system to that of an initially unoptimized and unsynchronized control, a fairer comparison than the first two experiments. Another purpose is to explore the capability of the system to correct a control regime to a situation to which it is not well-adapted. The improvement which is obtained in this case is much more significant. On the average, we obtain 2.5 times less collapsed lanes, less time wasted by a factor of 9.8 % and 104 % less cars waiting in the

the other are entry and exit points of the network. The intersections are linked by 15 streets with a total of 51 lanes. A network of this size is easily handled on a computer, but the interaction between the control decisions that can be made in each intersection is complex enough to make it difficult to control the system manually.

In this example, traffic lights are controlled using tables of fixed time plans for each intersection. Time plans for different intersections with the same indices are synchronized together. Only three different time plans exist in every intersection, which means that the possibilities for control are very limited, especially if synchronization is to be maintained. Our prototype therefore ignores the synchronization, which somtimes leads to non-optimal results but ensures a realistic test of the behavior of the algorithms.

As it is not feasible to test the system on the actual traffic network itself, we have based the experiments on a simulation whose statistics are those observed in a 20 minute period of the morning peak rush hour. The cycle time of the traffic lights is either 100 or 75 seconds. We start the simulation with an empty network.

Empirically, it was found to be best to run the reasoning system every 400 seconds of simulation time, which is about 4 to 6 times the cycle time of the traffic lights. The total number of possible solutions without constraints is $4^5 = 1024$ (5 controlled intersections, with 3 different "real" time-plans plus the dummy one). In general, the number of solutions found by the system varies between 30 and 100, except that it finds as many as 720 solutions for the initial empty network in which almost any control is possible. The method used to choose among different competing alternatives is to select the solution which makes the most appropriate changes for those streets in which traffic is closest to being collapsed.

We have made the following experiments:

1. simulation using the optimal traffic light controls chosen by traffic engineers, without adaptation by our control system.

2. simulation using the optimal traffic light controls, but with adaptation using our system.

3. simulation with randomly chosen control settings, without adaptation by our control system.

Figure 8: *The part of the road network of Lausanne on which we tested our method.*

Note finally that in this example, congested traffic strongly constrains the possible solutions, and in less constrained situations more solutions could exist. However, when the criterion for choosing among a variety of solutions is well-defined, they can be searched in that order so that the search can stop when the first one is found. For example, our prototype explores solutions ordered in increasing number of changes to the current settings. This again serves to limit the complexity of interpretation construction.

The complexity arguments of this section are corroborated by experience with a realistic example based on actual traffic data taken in the city of Lausanne, which we report in the next section.

4 Practical Example

In order to test the behavior of our system on a realistic example, we have applied it to a simulation of the traffic network shown in Figure 8, which models part of the street network of the city of Lausanne. It contains 16 intersections; 5 are traffic regulated and global interactions which directly define the environments, or by combination of labels in successive inferences. Neither case can arise in our system. All consumers are limited to single lanes or intersections and thus have strictly local effects. Furthermore, consumers always fire directly in response to external sensor data, so that their justifications cannot lead to combinations of environments established by other consumers.

The second process in which the complexity can explode is the construction of solutions based on the ATMS labeling. The complexity of this process depends on the number of ATMS labels and nogoods which have been generated, but more importantly on the total number of solutions that the labeling admits. While we cannot give a general formula for estimating the total number of solutions in our system, we can observe empirically that the large number of constraints strongly limits the possible choices. The constraints are imposed both by the choice sets as well as by the compatibility between flow and control models. Many of these nogoods are very general and, as shown in Figure 7, strongly prune the number of solutions that can exist at more detailed levels.

In the example of Figure 1, we have the following situation:

Assumptions/ Environment	Lattice Size	# Nogoods	# Environments
1	18	4	14
2	153	30	67
3	816	0	123
4	3060	0	63
5	8568	0	9
6	18564	0	0

In this example, each of the environments containing 5 assumptions defines one solution. The reason why there are 9 environments, but only four solutions (see section 2) is that 5 of the 9 environments contain unncessary `no-choice` assumptions which are eliminated during solution construction. Note that the constraints strongly prune the actual size of the ATMS lattice, thus keeping the complexity of the ATMS algorithms within reasonable bounds. Because of the local character of the constraints, we can observe a similarly strong pruning on larger examples. The limitation of the number of solutions means that combinatorial explosion during solution construction is also unlikely.

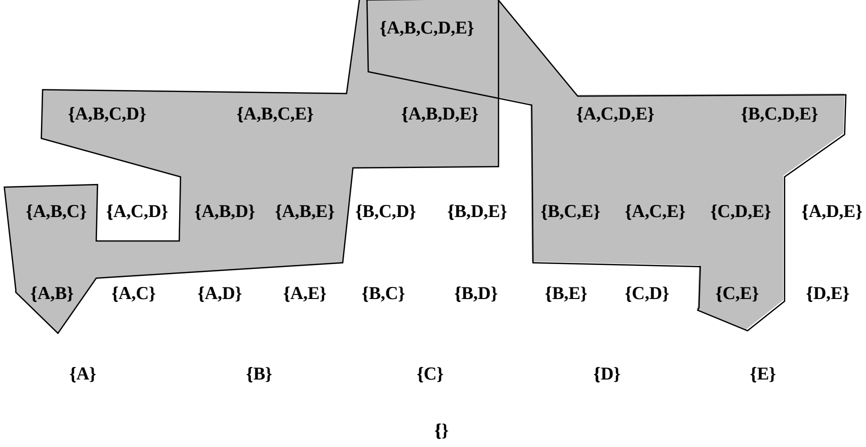

Figure 7: *Environments of assumptions can be arranged in a lattice of inheritance. Our system defines a large number of nogoods at the lowest levels of the lattice, which considerably reduce the search space. In this example, two nogoods with 2 assumptions leave only 4 out of the ten environments of 3 assumptions, and none with more than 3 assumptions.*

An ATMS encodes the variables of a problem as *assumptions*. Knowledge about their relation is expressed by rules which create *justifications* and *nogood justifications*, which in turn form *labels* of propositions. Combinatorial explosion can happen in two ways:

- the number of environments used in labels can explode, or
- the number of solutions in interpretation construction can explode.

As an example, we analyze the complexity of the example of Figure 1, considering only the three intersections and the two streets between them, which we assume to contain one lane each. We have a total of 18 assumptions reflecting the choices of the system:

- three different timeplans plus one `no-choice` option for each intersection are encoded by a total of 12 assumptions.
- three different flow models in the two lanes are modelled by another 6 assumptions.

We first address the number of environments forming the node labels. The key idea of the ATMS mechanism is to represent a large space of possibilities by minimal environments, taking advantage of their relation in an environment lattice as shown in Figure 7. The number of environments can explode when the minimal encoding of a label has to use many environments taken from the center of the lattice. This can arise either by

The case where there is *no solution* is characterized by the selection of the `no-choice` dummy timeplan at one of the intersections. The interpretation of the `no-choice` timeplan is that there is no available timeplan which would improve the situation, and only a suboptimal strategy can be generated. Each suboptimal strategy will leave the traffic situation with some particular problems. The choice of *which* suboptimal plan to generate should be made based on the problems one is most willing to accept. As each problem can be identified with a violated constraint, this means selecting constraints whose relaxation will lead to the generation of an acceptable strategy.

Consider a particular intersection for which only the `no-choice` control model remains as a feasible choice. The constraints which rule out the other alternatives are those whose associated ATMS node label contains a reference to the controls of the intersection. These are the candidates for relaxation.

Constraints in the candidate sets are relaxed in a sequence according to the following preference criteria:

- One can sort the lanes according to the severity of the local traffic situation based on the flow models, and first relax constraints associated with lanes whose flow models are farthest from congestion.

- For each lane, one defines a priority factor which defines the importance of non-collapsed traffic. This can be used to express the fact that some streets are used to "store" traffic on purpose, and collapsed traffic is acceptable for them.

The subset whose relaxation makes a solution feasible is found by suspending contraints in the order of preference until the intersection of the node labels contains a feasible solution for the control at the particular intersection.

3.4 Computational Complexity

An ATMS solves a problem by implicitly performing a complete search of the space of potential solutions. This can cause severe problems of computational complexity when this space gets large. The practical experience with our prototype has shown that the space of solutions and thus the computational complexity is manageable. In this section, we analyze the reasons for this well-behavedness.

choices for timeplans and flow models. A consistent strategy is a combination of assumptions which satisfies all the constraints and is not itself contradictory. It is computed from the node labels on the constraints by *interpretation construction* ([7]). This process intersects the sets of assumption combinations defined by the node labels, and subtracts the sets marked as contradictory in the label of the `*nogood*` node.

3.3 Choice of a strategy

The interpretation of the ATMS structures can leave us with three kinds of solutions:

- *One solution* is the ideal case. In such circumstances, the problem is well constrained and the strategy can be used as computed.
- *Many solutions* is the result of an underconstrained situation. One solution is chosen according to criteria given below.
- *No solutions* is the result of an overconstrained situation. In our implementation, this is the case when the "dummy time plan" `no-choice` is selected in any of the intersections. A constraint relaxation process is necessary to find an admissible solution.

In the case of *many solutions*, our prototype selects the solution which requires the least changes, since changing traffic light timeplans requires going through lengthy transitory regimes and can cause additional problems. Other criteria which could be used are:

- pick the solution which maintains as much synchronization among traffic lights as possible.
- classify the lanes according to severity of traffic, and pick the solution with the most appropriate traffic light settings for the lanes where traffic is closest to being collapsed.
- The criteria for declaring flow models collapsed (and unacceptable) can be weakened, resulting in additional constraints on the solutions.

on the measurements of the sensors. Constraints on consistency with the situation in adjacent roads could also be used.

- for each traffic light, the *control model* is a choice set among all possible timeplans for the traffic light. As explained earlier, for compatibility with current traffic control systems, our prototype requires that all lights in an intersection follow a fixed timeplan. Thus, all traffic lights in an intersection are controlled by a single choice set of timeplans and the `no-choice` model.

Note that choice sets could also be used for the topological models themselves, allowing the modeling of temporary modification of the street characteristics, for example by improperly parked vehicles.

As described in the preceding section, the flow model in each lane imposes restrictions on the choice of time plan in upward and downward intersections. These constraints are represented as `*nogood*` justifications installed by consumers attached to the flow model nodes for each lane. Once a flow model is assumed, the attached consumer is fired, installing the `*nogood*` justifications. In our description, only two types of such consumers exist, as the *fluid* flow model does not impose any constraints on the control model.

The *heavy traffic* consumer (attached to each node representing the heavy flow model) allows only time plan combinations that diminish the occupancy rate in the lane. This is achieved by constructing all the possible time plan combinations and discarding the ones having a smaller "green time difference", where the green time difference is the difference between the proportion of green time allocated to the outflow and to the inflow of the considered lane. When the contribution for the inflow comes from several lanes with different green time proportions, an approximation weighted by the actual charge of these different contributions is used.

The *collapsed* traffic consumer acts in a more drastic way. It allows only time-plans that give less green time at the upstream and more green time at the downstream intersection of the lane which the flow model is associated with.

As the consumers are fired, the label update algorithms of the ATMS computes a node label for each constraint that has been fired. The node label gives the set of assumptions which are consistent with the constraint. In our case, the assumptions are the different

Figure 6: *In the case of several successive streets with collapsed traffic, keeping a constant fixed control quickly results in a state where all streets are completely collapsed (bottom left). With model-based control, traffic is made fluid again in successive streets (bottom right).*

models as choice sets ensures completeness of the solution.

Constraints are encoded as consumers ([7]) attached to the nodes representing the choice sets. When constraints allow direct calculation of consequences, the consumers correspond to justifications among nodes of choice sets. When constraints rule out incompatible models, they correspond to `*nogood*` justifications. The consumer architecture automatically propagates changes in the measured data to keep the possible models consistent with the data. In particular, the three different kinds of models are represented as follows:

- for each lane, the *flow model* is represented by a choice set of three nodes, one for each possible flow model (*fluid*, *heavy* or *collapsed*). The choice is constrained based

solution	$J1$	$J2$	$J3$
1	1	1	0
2	1	1	1
3	2	1	0
4	2	1	1

Table 3: Admissible control strategies for the example problem.

- if the congestion in $R2$ is more acceptable than that in $R1$, the constraint on $R2$ should be relaxed, resulting in a solution in which the green time at $J2$ is decreased.
- if there is no preference between the acceptability of the the congestions of $R1$ and $R2$, a good choice is to not change the setting at $J2$ at all.

Assuming that the example corresponds to the last case, the actual settings would be kept. Solutions are given as the *interpretations* of the ATMS structures, which are all possible combinations of values admitted by the labels. Each of the four strategies shown in table 3.1 is a feasible solution. Any of the solutions shown in the table will contribute to solving the current problems in agreement with the human expert's solution (Figure 5). In this simple example, the situation is underconstrained, and the system selects one strategy according to criteria discussed later in this paper. Assuming that the third strategy is selected, the resulting traffic behavior is shown in Figure 6. This example shows how simple principles can be combined to generate strategies of human experts. Note that our system currently only generates instantaneous decisions. A strategy emerges through coupling of the system with a traffic simulator. A future extension would be to incorporate qualitative prediction rules in the reasoning mechanism which asserts the constraints, so that consistency constraints are asserted to prevent any potential future problem.

3.2 Representing models in the ATMS

For each topological element of the traffic network, there is a *choice set* ([6]) of flow or control models which could be associated with it. A choice set is a structure in which the ATMS forces exactly one of the choices to be assumed at any given time. Encoding

intersection	*time-plans*
$J1$	1,2
$J2$	No Choice
$J3$	0,1

Table 2: Labeling of the choice sets found by the ATMS

for $J1$, 1 for $J2$ and 2 for $J3$, which means that the flow models associated with both roads quickly become of type *collapsed*. As there are constraints attached with each possible flow model (using the consumer architecture and `*nogood*` justifications), the choice of the time-plans in the upward and downward intersections of a lane with a collapsed flow type is limited according to the following constraints:

- Only timeplans giving less green time can be selected in the upward intersection.
- Only timeplans giving more green time can be selected in the downward intersection.

The propagation mechanism of the ATMS results in the different labels for each choice set shown in table 3.1. $J2$ has no admissible choice because traffic flow in $R1$ is of type *collapsed*, which means that more green time should be given in $J2$. On the other hand, flow in $R2$ is also of type *collapsed*, and consequently less green time should be given in $J2$. As the choice of timeplans in an intersection is exclusive, the two facts are contradictory and leave no possible timeplans which would correct the problems. In our system, this case is represented by the `no-choice` control model, which does not correspond to any physically realizable control, but means that only a suboptimal control is possible. The different suboptimal solutions are characterized by the traffic problems they leave unsolved, and must be selected according to a preference order of acceptability of these problems. This is the reason for using the iterative constraint relaxation procedure. In the example, there is a choice of solutions, corresponding to relaxing different constraints:

- if the congestion in $R1$ is more acceptable than that in $R2$, the constraint on $R1$ should be relaxed, resulting in a solution in which the green time at $J2$ is increased.

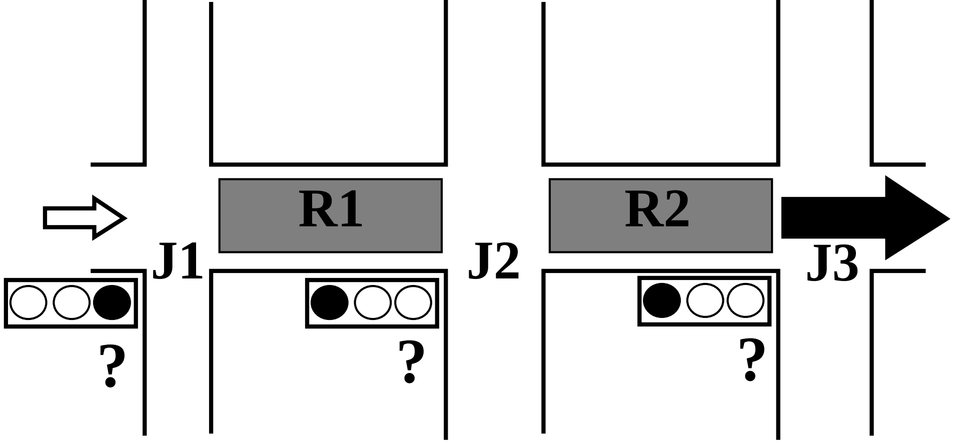

Figure 5: *The solution: diminish the inflow and augment the outflow.*

time-plan	*green-time*		
	$J1$	$J2$	$J3$
0	90 %	80 %	90 %
1	60 %	63 %	60 %
2	40 %	40 %	33 %

Table 1: Green time proportion for the different timeplans in the intersection

a traffic engineer would use to resolve this situation is to increase outflow from the last exit segment, and reduce the flow into the first segment, as shown in Figure 5. For the first and last intersections, there are control models which indeed improve the situation: give less green time to the first intersection, and more green time to the last. For the intermediate intersections, the constraints conflict and do not leave a solution that is consistent with the situation. A traffic engineer faced with this problem would decide that the problems at these intersections cannot be corrected immediately, and must be solved later. He will therefore leave their control regimes unchanged from the current settings.

We now see how this same strategy can be computed by combining the principles of our system. In the example, we have two roads, $R1$ and $R2$, and three intersections $J1$, $J2$ and $J3$. For the each intersection, we have three possible time plans whose green time proportions are given in table 3.1. Assume that initially, the time-plans are time-plan 0

3 Generation of Control Strategies

The main problem of traffic control is that the choice of *local* controls must take into account the *global* traffic situation. Controls must be chosen according to a global *strategy*, which we define as a combination of control models for all controlled intersections in the network. A strategy is *consistent* whenever the models satisfy all of the consistency constraints specified in the network. By enforcing global consistency, each of the controls in a strategy takes into account the traffic situation in other parts of the network.

Computing a strategy amounts to a *search* in the space of all consistent combinations of the different control models. A mechanism which is designed to efficiently search a space defined by constraints is the *assumption-based truth maintenance system* (ATMS, [5, 9]). Our prototype generates control strategies in the following steps:

1. Based on the topological model and sensor readings, instantiate the set of candidate choices for flow and control models.
2. Successively install the consistency constraints as ATMS justifications by firing the consumers defining them.
3. Find all consistent strategies as interpretations of the resulting label structure.
4. If more than one solution remains, choose among them to find the best one. If the best solution contains `no-choice` control models, find the corresponding constraints, select those which should be relaxed, and restart the process from step 1.
5. Choose the control models according to the final strategy.

This process is similar to that of model-based diagnosis with fault models ([3, 16, 4]), where the diagnosis is found as the combination of models which is consistent with the observations. One important difference is that in synthesizing a control strategy, unknown faults (`no-choice` control models) cannot be allowed. An acceptable strategy can be found by iterative relaxation of constraints, as explained later in this section.

3.1 Clearing a sequence of congested streets

To illustrate the process of computing a strategy, consider the example of a completely blocked street with several successive intersections shown in Figure 1. The strategy that full-fledged system, where intersection controls are dynamically adjustable, the synchronization between controls is important and could be modelled by explicit *synchronization models*.

2.4 Constraints

The fact that not all combinations of topological, flow and control models are compatible is modelled by *constraints*. Physical constraints represent the laws of traffic behavior, such as the fact that a control regime limits the charge exiting from a lane. Examples of such constraints are:

- traffic flow can not be fluid and collapsed at the same time.
- two different time plans can not be selected at the same time.
- constraints on the control strategy combinations when they are connected by synchronizers.
- flow models must be consistent with charge/occupancy rate measurements.

Besides the physical constraints, there are *compatibility* constraints between control models and the traffic situation, which encode a traffic engineer's problem solving knowledge. To avoid developing *collapsed* traffic, no control strategy which can transform a *heavy* flow model into a *collapsed* one is allowed. When collapsed traffic already exists, the compatibility constraints allow only control regimes which will reduce the congestion in the future. The compatibility constraints incorporate a short-term prediction of the traffic evolution. Better performance might be achievable by long-term prediction, this is discussed later in the paper.

Any combination of models and sensor readings which is consistent with these constraints corresponds to a possible actual traffic state. Note that this traffic state may not be uniquely determined in the absence of complete measurements. Also, accidents and other unforeseen events which perturb the traffic network could be detected by diagnosis when no consistent model is found.

2.3 Control Models

The control model describes the strategies for controlling the lights at individual intersections and their synchronization. In current traffic control systems, traffic lights are controlled by fixed *timeplans*. A time plan is a sequence of states of the traffic lights at the intersection. It is characterized by the *split* (percentage of green time) for each direction of an intersection and the *cycle time*. For each intersection, there is a library of precomputed time plans. Time plans can be synchronized between intersections, but this is only implicit in compatibility constraints between timeplan choices.

Following the analogy with model-based diagnosis, control models correspond to the different operating modes of devices, with the conditions for their applicability defined implicitly by the compatibility constraints. Any selected control model which is not working under its operating conditions is a fault, and the models which fit the compatibility constraints indicate a better control which would improve the situation. A dummy control model, `no-choice`, is used to represent the case analogous to an unknown fault. Its interpretation is that there does not exist a control model which is compatible with the constraints, which indicates that contraints must be relaxed to find a suitable control model.

More powerful control would be possible if the timeplans themselves were dynamically adjusted to the traffic situation based on the characteristics of the intersections. This type of control allows more flexibility for adaptation, but its practical implementation poses several difficulties. First, the quantities involved are continuous and can not be set by search in a finite domain. This problem could probably be addressed adapting techniques developed for temporal reasoning ([10]). Second, varying split and cycle time requires passing through transient states which can take significant amounts of time and whose effects on traffic flow are difficult to predict. Because of these difficulties and the absence of suitable global control systems, practical systems are based on fixed timeplans. In order to be able to test our system on realistic data, our prototype so far is based only on the fixed timeplan model.

In an intersection, each direction which is controlled by a traffic light is associated with a separate control model. In the current prototype, synchronization is not considered, as the fixed timeplans would then allow only very little variation of controls. In a

Figure 4: *The three different types of flow models: fluid, heavy and collapsed traffic.*

rate builds up, and eventually the model changes to heavy traffic.

- high occupancy rate, high charge: *heavy* traffic, but still flowing. This situation is maintained as long as traffic control is synchronized at the average speed. Otherwise, the situation degenerates into collapsed traffic. When the charge is reduced, the model changes to fluid traffic.
- high occupancy rate, low charge: *collapsed* traffic. Typically, traffic can not move accross intersections because the exit lanes are constantly occupied. Collapsed traffic can change to fluid traffic when the incoming charge diminishes or the outgoing charge increases, but never directly to heavy traffic.

A flow model is associated with each lane of the topological model. With the passage of time and depending on the control models, flow models are propagated to adjacent elements.

The *capacity* of a lane - the maximum charge it can carry - is equal to the maximum flow rate of heavy traffic when the *fluid* and *heavy* flow models apply, and equal to the (much lower) flow of the *collapsed* traffic when this flow model is applicable. For fluid and heavy flow models, total traffic flow is further limited by the control model at the lane exit, and for collapsed traffic capacity limits carry over from the downstream lanes.

characteristics of traffic flow in it. This is further discussed in the section on flow models.

Intersections are modelled as a collection of *directions*, models which connect a specific *entry lane* and an *exit lane*. Two directions are said to be *compatible* if traffic flow can occur simultaneously in both of them, and this is modelled by explicit constraints. When traffic is congested across directions which are not compatible, they can block each other and spread the congestion.

2.2 Flow Models

In traffic control, the fact that car movement has an *inertia* means that the characteristics of traffic propagate between different streets. For example, congested traffic in an incoming street will continue as congested traffic in the devices connected to it, rather than becoming fluid. It is therefore useful to represent the instantaneous state of traffic by explicit *flow* models.

At each point, traffic flow is characterized by two quantities: *charge* and *occupancy rate*. The charge is the flow of traffic through the point, expressed in number of vehicles per time unit. The occupancy rate is the proportion of time that the point is occupied by a vehicle. The two quantities are measured by traffic sensors for each lane. Note that the average speed in a lane is proportional to charge/occupancy rate.

A lane is logically divided in two parts, the first corresponding to the "free" part of the lane, the second to the queue formed by the cars waiting at the end of the lane. The boundary between them moves dynamically according to the traffic in the lane and the traffic light settings at the end of the lane. Because a sensor only measures the characteristics of one part of the regions, there are often two sensors per lane, one at its entry and one at its exit. In our current system, we are interested only in the behavior of traffic in the free part of the lane, although more complex strategies might also take into account whether or not the queue at the light clears. Based on the combinations of sensor readings at the lane entry, we distinguish three different qualitative types of flow models, each corresponding to a different behavior of traffic ([1]) and illustrated in Figure 4:

- low occupancy rate: *fluid* traffic conditions where the charge is a continuous parameter. This is the ideal situation, and usually no changes in the control strategy are needed. If the amount of traffic grows and charge exceeds the capacity, occupancy

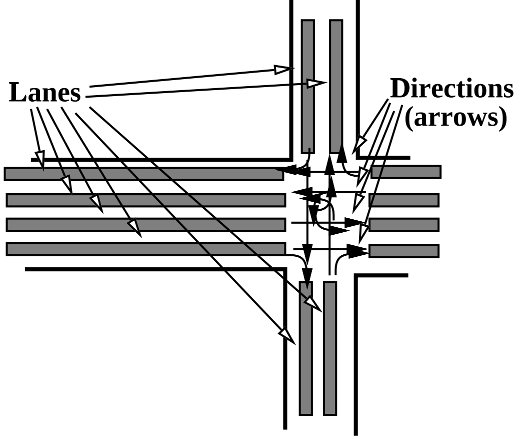

Figure 3: *Part of the topological model for a small road network.*

The influences between models are described by constraints on their consistency. A typical consistency constraint is that the traffic flow in one street will be found distributed in the following streets at the next time instant. Consistency constraints on control models express their compatibility with the current traffic situation. For example, they rule out control models which add to existing congestions, or which give useless green time which could be more profitably used.

2.1 Topological Models

Topological models represent streets and intersections as a network of devices, as shown in Figure 3. Associated with each device are its fixed parameters, such as the length of a road or the topology of an intersection, which are required for predicting the evolution of traffic flow. Usually, topological models do not change in time, but events like road work, accidents or severe weather conditions can require changes.

Topological model represent two kinds of objects: *streets* and *intersections*. Streets are modelled as a collection of *lanes*, which are devices that can carry a sequence of vehicles in one direction. Two important assumptions about streets are that the number of lanes cannot vary along its length, and that the net "inter-lane traffic" in a street is balanced. A car which enters a street in a specific lane also leaves the street using the same lane. A lane is characterized by two parameters, its length and the average speed of cars under ideal conditions. Note that the *capacity* of a street - the maximum number of vehicles per time unit that can pass through the street in a given time unit - depends on the even banking. We expect our techniques to be as successful in such domains as they are in traffic control.

This paper is structured as follows. In Section 2, we define the models and compatibility constraints between them. In Section 3, we define the algorithm and illustrate it on the example of Figure 1. Finally, we describe the results of a larger simulation based on actually measured traffic data.

2 Traffic Models and Constraints

Most work in model-based diagnosis is based on *device-oriented* models of the system to be diagnosed. Such a model consists of a network of *devices* which are connected by *conduits*. Each device has a number of quantities which follow device *laws*, and faults of a device are modelled as violated device laws. Conduits identify shared parameters between devices.

In a traffic network, the devices are streets and intersections, and the conduits are their interconnections. In electronic circuits, the state of devices is modelled by continuous parameters. In traffic control, it is more convenient to model the state by the selection of explicit models of traffic flow and control regimes. There are thus three different kinds of models in our system:

- a *topological* model which defines streets and intersections as devices with fixed characteristics,
- *flow* models for the characteristics of units of traffic that propagate through the network, and
- *control* models for individual intersections and the synchronization between them.

The state of traffic in a street is represented by its associated flow model, and the control of traffic lights in an intersection by the associated control models. The temporal behavior of the system can be qualitatively modelled by flow models propagating through topological models influenced by laws defined in control models. The current prototype does not incorporate long-term prediction, and only local propagation to adjacent devices is modelled.

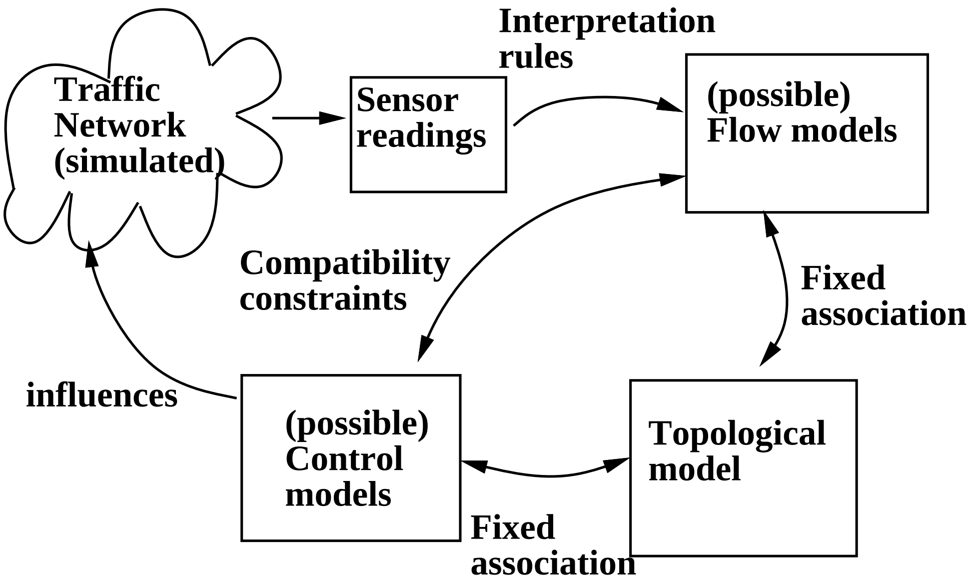

Figure 2: *Structure of the prototype system.*

control system integrates diagnosis with correction in an elegant way: each diagnosis is a control strategy consisting of a set of fault models which index corrections to the control, and a set of faults of unknown type which cannot be corrected.

For selecting the best control strategy in this set, the most important criterion is the severity of the problems which the strategy does not correct, i.e. the severity of the faults of unknown type. In general, this selection can not be made directly on the basis of the faults themselves, but should consider the traffic problems they cause. This requires explicitly suspending some of the compatibility constraints underlying the discrepancies until all problems are corrected by the resulting strategy. We have implemented the techniques in a prototype whose structure is shown in Figure 2. Since it is not feasible to experiment with our system on actual traffic situations, traffic flow is modelled by a simulator which follows statistics taken from actual traffic data. The simulation defines sensor readings which activate the applicable flow models. The system periodically recomputes the set of consistent control models, which generate the control events for the traffic simulator.

Our work points out an extension of model-based diagnosis to model-based control. The synthesis of globally consistent control strategies is an important problem in many complex systems, such as the control of communication networks or industrial plants, or in the street. Each intersection is associated with a *control* model which describes the way traffic lights at the intersection are regulated. A suitable control regime is one in which the control models are *compatible* both with each other and with the topological and flow models. The notion of compatibility, rather than optimality, makes it possible to evaluate control regimes with respect to extreme situations. For example, only controls with little incoming green time are compatible with an intersection whose outgoing lanes are blocked. However, other settings will have about the same effect on the instantaneous traffic situation, so that optimality criteria could not distinguish between them.

The basic idea of model-based diagnosis is that faults manifest themselves as discrepancies between the model of correct behavior and the actual observations. Computation of a diagnosis is a *search* for a combination of models of faulty components which explains all discrepancies. In control problems, the dynamic behavior complicates the dependencies between control regimes and system behavior, so that it is difficult to compute discrepancies between *behavior* and control regimes from first principles. A more pragmatic approach we adopt in this paper is to define explicit *compatibility constraints* between behavior and control regimes. The result of applying the model-based diagnosis technique to the violations of these constraints are combinations of control models which should be changed to potentially correct the problems with the traffic situation.

However, the task is not just to *identify* control models to change, but also to find better controls which solve the problems in the current situation. This can be accomplished by not just computing which control models are faulty, but also the *way* in which they are faulty, and using this information to select better strategies. Our system does this by modelling faulty controls by those control models which would be more suitable for the particular situation: the model of the fault is also the model of its correction. For example, an unadapted control strategy may be that of a traffic light which allows cars to enter an intersection event though the exits are completely blocked, as in the case of Figure 1. The fault is modelled by any control model which gives less green time to enter the intersection. Situations where the requirements conflict and no suitable control exists are modelled as an unknown fault.

Research on GDE+ ([16]) and SHERLOCK ([4]) has shown how to find a diagnosis as a combination of fault models and unknown faults. Using this method, our traffic

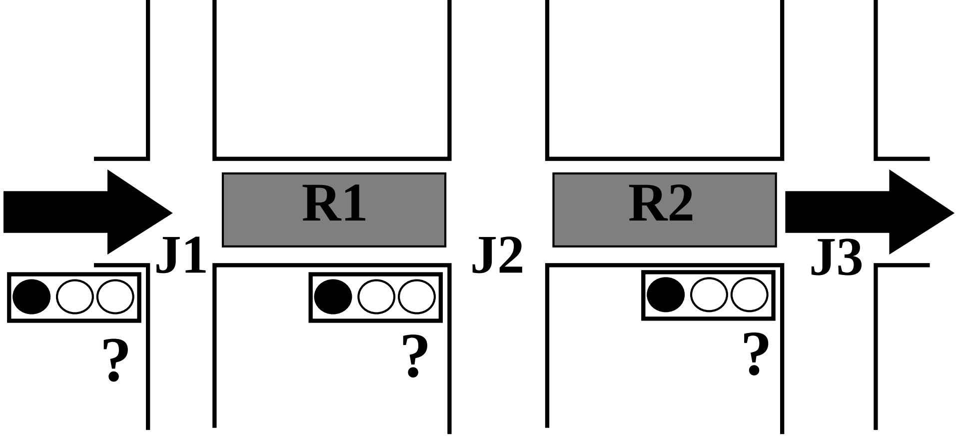

Figure 1: *A situation in which several successive streets are completely blocked by standing traffic.*

suggest plans to resolve the typical congestions. SAGE is very successful in resolving critical situations by suggesting drastic strategy changes. However, the rule-based approach can not be applied to networks whose behavior is not completely known to experts, and might miss solutions to unprecedented situations.

In this paper, we show how model-based reasoning can replace the heuristic rules by first-principles reasoning based on a model of traffic. Problem solving in traffic control consists of two aspects:

- **diagnosis** of the current traffic state to find those control regimes which are ill-suited, and
- **correction** of the control elements found by diagnosis.

An important characteristic of traffic control is that problems can in general not be resolved *locally*, but require a globally consistent strategy. This characteristic is shared with electronic circuits, where the strong interactions between components make local diagnosis equally difficult. In the domain of circuits, these difficulties have led to the proposal of *model-based* diagnosis, first defined by Reiter ([14]) and DeKleer and Williams ([3]). In this paper, we show how to use this approach for not only diagnosing traffic control problems, but also generating the control strategies which help to correct them.

We model the traffic system as a network of devices: streets and intersections. Each street is associated with a *flow* model which describes the characteristics of traffic flow

1 Traffic Control Problems

Automatic urban traffic control is important to relieve large cities of their serious congestion problems. As control of individual vehicle movements is impossible (at least in the short term), traffic flow can only be regulated by controlling traffic lights based on the interpretation of sensors. Currently, most traffic control systems follow timeplans taken from a preprogrammed set and chosen by human traffic engineers based on their assessment of the traffic situation. Occasionally, the timeplans include *local* sensory control, for example giving green only if cars are actually waiting for it.

Timeplans are optimized for a set of common traffic situations, such as morning and evening rush hours. In exceptional situations, such as an accident at a crucial intersection, the controllers must improvise. In many cases, the complexity of interactions created by the road network means that they are unable to find good solutions in real time. Computer support for this situation would be highly desirable.

There exist several systems for automatic traffic control which are in actual use. The most successful system is SCOOT ([13]), in which the parameters of the traffic control strategy are dynamically adjusted based on information from sensors and a continuous traffic model. SCOOT is used in several British cities and has been very successful for optimizing traffic flow in *non-congested* situtations. However, resolving severe congestion situations requires more drastic changes in control policy than can be expressed by the parameter variations allowed in SCOOT.

As an example, consider the situation shown in Figure 1. Since no more cars can enter the blocked streets, letting additional cars enter at J_1 might block this intersection, causing the problem to spread further in the network. The solution is to temporarily block entry to the congested area while creating an outlet for the congestion. Only when the congestion has cleared up sufficiently can additional cars be allowed to enter. In an optimization policy, the many minor problems created by blocking an entry often outweigh the one major advantage gained by clearing the congestion: it is stuck at a local minimum. Strategies for overcoming the local minima can be derived more easily by reasoning on an explicit *model* of the traffic network.

This fact has been the motivation for the SAGE system ([15]), a rule-based expert system used to resolve congestion situations in the center of Paris. In SAGE, specific rules

Summary

Urban traffic control is a difficult problem because of the complex interdepence of control decisions. Known techniques for achieving global control are based on parameter optimization techniques or heuristic expert systems. Optimization fails in severely congested traffic situtations which require a change in global strategy. Heuristic expert systems require knowledge of all possible traffic situations, which is difficult to obtain, especially when construction and incidents cause frequent changes to the traffic network.

In this paper, we show how the techniques of model-based diagnosis can be used to select coordinated control plans for networked systems of this kind. Suitable local control strategies are those whose underlying assumptions are *consistent* with other control strategies, the state of the road network, and traffic flow. We describe a system which uses an assumption-based truth maintenance system (ATMS) to compute suitable strategies. The system has been tested both on synthetic examples and on simulations using actual data, and results are encouraging.

Model-based Traffic Control

Eric Sauthier and **Boi Faltings**
Laboratoire d'Intelligence Artificielle
Ecole Polytechnique Fédérale de Lausanne
MA-Ecublens, 1015 Lausanne, Switzerland
e-mail: faltings@eldi.epfl.ch

December 19, 1991
Keywords: Model-based reasoning, model-based diagnosis, ATMS,
traffic control