A METHODOLOGY TO ESTIMATE POSSIBLE GAINS FROM REAL TIME CHANGES IN ROAD NETWORK TOPOLOGY

Rute Geraldes, Instituto Superior Técnico (IST), rutecg@civil.ist.utl.pt

José Manuel Viegas, Instituto Superior Técnico (IST), viegas@civil.ist.utl.pt

ABSTRACT

Assuming the possibility of topological changes (changes in street's ways of traffic flow and turning's prohibitions) in a road network in real time and the hypothesis that such changes can be beneficial to manage traffic incidents, a methodology was created to estimate the possible gains deriving from this approach.

In order to estimate the gains, two major advances are needed. We need to find out the best (or at least a rather good) network topology to bring quick relief for each incident situation, which in turn requires that we can evaluate different topologies based on the effects of those changes over the network. It can be seen as bringing together an outer process that searches for the optimal topology and an inner process to evaluate each topological configuration.

In this paper we describe both step by step. The outer process is supported on Matlab multi-objective genetic algorithm toolbox with custom fitness, creation, mutation and crossover functions, programmed for this specific purpose. The inner process is supported on Aimsun's microsimulator, dynamic traffic assignment model and traffic management actions, along with some extra scripts developed specifically for this analysis.

Basically the fitness function in the Matlab genetic algorithm calls Aimsun, or more precisely, calls an Aimsun script that loads a given network and executes all the steps needed to simulate an incident with consequent topological changes as a response to it (given as inputs) and to evaluate that network *performance.*

Keywords: Traffic Incidents; Urban Road Congestion; Network Topology Optimization; Micro Simulation; Multi-Objective Genetic Algorithms

INTRODUCTION

This presentation, as well as the one entitled "Real time changes of road network topology as an approach to mitigate the effects of incidents in urban congestion" (Geraldes & Viegas,

2010), results from work done for the PhD Thesis in Transportation of the first author under the theme "Mitigation of road congestion problems caused by incidents".
Briefly, as explained in more detail in (Geraldes & Viegas, 2010), we assume the possibility of topological changes (changes in street's ways and turning's prohibitions) in a road network in real time (based on smart signaling devices) and the hypothesis that such changes can be beneficial to manage traffic incidents impacts.
Here we describe the methodology created and used to estimate the possible gains deriving from applying this new approach and try to confirm our hypothesis.

Situations to compare

When talking about gains we are always referring to gains from one situation against another one, while attempting to achieve an ideal situation.
In this case we need a methodology to evaluate a road network under four circumstances, normal working (no incidents), the same network with an incident, the same network with incident and an advanced traffic information systems to help managing traffic under incident situations (a very studied solution with several real world applications) and finally our approach, the same network with an incident, an advanced traffic information system (ATIS) and network topological changes as response to the incident. With these four evaluations it is then possible to compare the gains of the fourth situation against situations two and three.

MEASURING GAINS

Our aim involves evaluating a road network under a new system that changes the network topology. That system doesn't exist yet. Since there aren't real implementations allowing real measurements of its performance, it is clear that previous evaluations and comparisons have to be made under simulated situations, recurring to a model of the network and to traffic assignment models to reproduce drivers' behaviours under the different circumstances.
Looking for a common platform (so that evaluations could be comparable) to simulate the described 4 situations and taking into account their specificities, it became clear that the more traditional traffic assignment approaches could not do the job and there was the need to use a Microscopic Dynamic Traffic Assignment Model.
Popular and largely used Equilibrium models, well summarized by Yosef Sheffi in (Sheffi, 1985) clearly do not apply because the occurrence of an unexpected incident in a network does precisely introduce a disequilibrium in the network.
Other macroscopic static assignment models available in transportation modelling literature like (Ortúzar & Willumsen, 2001) also do not apply.
An incident introduces a sudden and discontinuous change in the capacity of at least an element of the network, also allowing topological changes on the network, even if the demand could be well defined by a static OD matrix, the supply side (the network) is dynamic, which requires the introduction of the variable time in the problem.
Macroscopic models are not able to model properly congested urban networks even without incidents, because of their difficulty in incorporating delays in nodes (some times more important than delays in links on urban congested networks) and links interactions, also

important and common in congested networks with short links like the urban ones. In modeling the effect of an incident in a network it is important to replicate the queue propagation effect that the incident introduces (how a queue propagates backwards from the link with the incident) and we are expectedly dealing with a network under heavy congestion.
Under dynamic traffic assignment models there is a wide variety of models that can be divided in two large groups: analytical (Mathematical programming, Optimal Control, Variational Inequality) and simulation-based approaches. In (Peeta & Ziliaskopoulos, 2001) a review is made of different approaches to Dynamic Traffic Assignment and our interpretation of it is that, while analytical approaches are still struggling with various deficiencies and fighting to be the one universally accepted, simulation based approaches are gaining popularity because they address, with relative ease, several modelling issues that are troublesome in analytical formulations.
There are actually several implemented and commercial simulators applying simulation based DTA models (some examples: Aimsun, Contram, DynaMIT/MITSIMLab, Dynasmart).
Our choice has been Aimsun mainly because it was a software that our research group already had, it is simultaneously user friendly for the tasks it already offers and adjustable/extensible/programmable for new utilizations, it works in an integrated way with macro and micro modeling allowing to start with macro data (more commonly available then micro) and a large network and progressively refine our focus, infer data for smaller areas and work on a micro scale and finally because by being the simulation software utilized in the European project PRIME-Prediction of Congestion And Incidents In Real Time, For Intelligent Incident Management And Emergency Traffic Management (Information Society Technology Programme) and by incorporating traffic management actions, tools and issues, like incidents, it made us believe it would have the necessary tools to model our incident situations.

Modelling the different situations to compare

In synthesis, there is the need to model 4 different situations already explained:

- Situation 1 – Normal conditions (without incident)
- Situation 2 – Conditions under an incident situation
- Situation 3 – Conditions under an incident situation and ATIS
- Situation 4 – Conditions under an incident situation and ATIS plus topological changes in the network

First, input data required by Aimsun regarding network layout, traffic demand and traffic control must be introduced.
Aimsun has two dynamic traffic assignment algorithms, the dynamic traffic assignment (DTA) and the dynamic user equilibrium (DUE).

> "A myopic implementation of a heuristic stochastic DTA in Aimsun, based on the above considerations, is the following:
>
> The simulation process based on time dependent routes consists of the following steps:

> *Repeat until all the demand has been assigned:*
>
> *1. Calculate initial shortest routes for each O/D pair using the defined initial costs.*
>
> *2. Simulate for a user defined time interval assigning to the available routes the fraction of the trips between each O/D pair for that time interval according to the selected route choice model and obtain new average link travel times as a result of the simulation.*
>
> *3. Recalculate shortest routes, taking into account the current average link travel times.*
>
> *4. If there are guided vehicles, or variable message signs suggesting rerouting, provide the information calculated in 3 to the drivers that are dynamically allowed to reroute.*
>
> *5. Go to step 2.*
>
> *(...)*
>
> For a DTA to become a DUE the behavioural assumptions on how travellers choose the routes have to be consistent with the dynamic user equilibrium principle. Ran and Boyce 1996, formulate the dynamic version of Wardrop's user equilibrium in the following terms: If, for each OD pair at each instant of time, the actual travel times experienced by travelers departing at the same time are equal and minimal, the dynamic traffic flow over the network is in a travel-time-based dynamic user equilibrium (DUE) state.
>
> (...)
>
> This process implements an iterative process where minimize, applying the method of the successive averages (MSA), the function of Relative gap that represents the achievement of the user equilibrium." (TSS - Transport Simulation systems, April 2010)

Although DUE could be used to model situation1 (day-by-day equilibrium), as already discussed it could not apply to situation 2. DUE would only be applicable to situation3 under an ideal perfect information system that would be able to instantaneously predict drivers' paths choices and predict paths costs and instantaneously transmit them to the drivers that would instantaneously readjust their routes according to the transmitted path costs (within day equilibrium). The same applies to situation 4, further for modelling situation 4 we need Aimsun traffic management actions that are only available with DTA algorithm.
So for Situation 2, 3 and 4 DTA has to be, for situation 1 it would only be DUE if we could guarantee initial conditions in the network for DUE and DTA in order that results in the different situations could be comparable. While Aimsun DTA allows us to save and utilize an

initial network state for simulations, DUE doesn't and some small tests showed that with the same OD Matrix the warm up of DUE and the warm up of DTA leave the network in pretty different initial states for the simulations.

But when using DTA it is possible to set it to use paths and route choice calculations from previous simulations instead of recalculating it all over again. This means it is possible to run a simulation with DUE, store its path assignment results and then start a simulation with DTA with a requirement to follow those results.

So we have decided to run a DUE simulation until the instant of the incident, under normal conditions, as a kind of warm up to load the network, and save its path assignment results (before incident DUE path assignment results). Then if we run a DTA forcing 100% of the vehicles to follow the path assignment results from previous simulation and save the network state at the end of the simulation, we have a good approximation for a network initial state to be used by all the simulations trying to reproduce our 4 described situations (Initial State 1 – Incident Instant).

Then we assumed that without information, either with or without an incident, drivers use the paths that they use every day, that conduct to a day by day equilibrium and that can be found by storing the results from an DUE simulation for the period since the incident occurs until the instant that, without doing anything to manage it, the effects of the incident are no longer felt (during incident DUE path assignment results).

Situation1 can be modelled by simulations departing from initial state 1, for the period of time during which the incident effects are felt, with a DTA that forces 100% of the vehicles to follow during incident DUE path assignment results.

Situation 2 can be modelled with the same premises as situation 1 plus the introduction of an Aimsun incident traffic action for the place and the duration of the incident.

Situation 3 can be modelled as situation 2 for a short period of time while incident has not been detected yet. Then, once the incident is detected the ATIS gets in action and drivers no longer take their usual paths but the ones resulting from a C-Logit Route choice model taking into account path costs actualized every 'x' seconds by the ATIS. Paths are evaluated and chosen by drivers not only when they enter the network but also at any time when they receive new information on path costs, making possible to watch drivers changing paths in the middle of their trip. In Aimsun this correspond to a DTA with 100% of the cars following C-Logit route choice model, dynamic enroute, a cycle of 'x' and keeping activated (while the incident lasts) the incident traffic action referred for situation 2.

Both periods before and after incident detection cannot be covered by only one simulation in Aimsun but a simulation of situation 2, representative of the mean performance, can be run in order to save a traffic state in the instant of the incident detection. Such traffic state can then be introduced as an initial state when modelling situation 3 after incident detection.

What has been said to situation 3 can be used as a base for modelling situation4, but now there is also the need to introduce topological changes in the network.

Topological changes mean changing a street/section direction and consequent turns in the extreme nodes or just changing permissions/prohibitions of turns already physically possible without changing sections directions.

In order to model this in Aimsun sections in which directions might be changed must be duplicated (one for each direction) and all possible turns must be designed in changeable nodes. Then using Aimsun closing turn actions it is possible to control which turns and sections are used in each instant.

An API must be added to Aimsun scenario to model situation4 after incident detection. On that API close turn actions must be added for:

- the entire simulation, for turns that must not be used either before or after incident detection or turns that are supposed to be closed when the incident is detected and do not leave from section which direction is about to be changed.
- The time period necessary to empty sections which directions are about to be changed, for the turns that are supposed to be open when the incident is detected but imply a section direction change first.
- The time period since sections which directions are about to be changed are empty until the end of the simulation, for the turns leaving from sections which directions are changed.

The time necessary to empty sections before changing their direction is estimated based on past history and/or traffic control plan cycles.

Since this is an estimated time and covering any possibility that the section is not already empty when turns leaving it are closed, a force turning action is also activated for the u-turn leaving section with former direction to section with new direction at the instant when directions are changed. With this scheme no vehicle gets stuck or lost in the network.

In order for results to be significant, several replications must be run for each situation. A good number of replications could be the capable of guarantee a maximum error of 3% in the performance measure with a confidence interval of 95%.

Performance Measure

The network performance is measured in terms of travel time per km. Every time a car leaves a section the time it took inside the section as well as the distance it travelled inside the same section is registered. If we sum the travel times of all the vehicles leaving the section during a period of time, sum the distances travelled for all cars leaving the section in the same period and then divide the first sum by the second we have the travel time per km for that section and that time period.

If we make the average of section travel times per km for a given period, for all network sections we get a travel time per km for the entire network for the same period. This average is not a simple average, instead it is weighted by 3 factors:

- The length of the section, because a bigger section represents a bigger part of the network;
- Period of time of the total simulation period in which the section is open, in the case of sections representing one of the directions of a street which direction was changed its travel time per km is only taken in account in the proportion of the time during which the street has that direction;
- A function of the distance to the incident. Sections closer to the incident in paths accessing the area of the incident are the most affected and the ones whose performance changes more in consequence of the incident. Valuing equally the sections of the network that are almost not affected by the incident and the ones under the incident influence zone might make the network performance measure

too insensitive to the incident and consequent topological changes impacts. So, an inverted logistic function of the distance to the point of the incident, similar to the one used to measure accessibility (Kocks Consult GmbH, 1978), was used.

A first calibration of the formula was done taken into account that the sections to be more relevant are the ones near the incident, and in this first calibration "near" was considered to be within 600m of distance with a relative importance of 90% and "far" further then 1800m (3 times 600m) of distance with a relative importance of 10%. But other definitions of near and far can be tested.

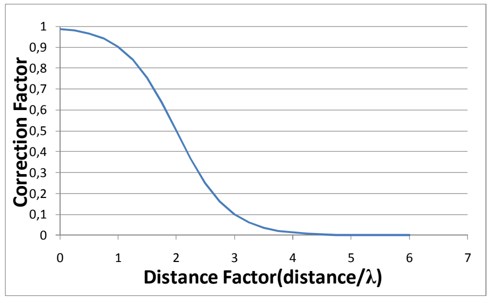

Figure 1 – Inverted logistic function

NETWORK TOPOLOGY CHANGES TO CONSIDER

The problem now is what network topology changes should be introduced in the 4th situation. To be fair and in order to infer if our base hypothesis is true they should be optimal or at least good changes.

So we are dealing with a combinatorial optimization problem in order to find out which combination of closing/opening turns actions lead to the best network state in terms of our performance measure.

The hypotheses are too many in order to allow exhaustive testing, even if it is in a smart way. Our problem can actually be seen as a kind of a network design problem or a network optimization problem, that result most of the times in NP-complete combinatorial problems and most of the times are solved recurring to metaheuristics. (Gen & Cheng, Genetic algorithms & engineering optimization, 2000)

In the present case the objective function of the optimization problem come from a simulation evaluation done in Aimsun, thus it is a not differentiable noisy function.

Genetic algorithms because of its stochastic nature have a particular potential as a tool for optimization when the evaluation function is noisy. (Fitzpatrick & Grefenstette, 1988)

Normal genetic algorithms deal only with one goal/objective, but in this case one can be insufficient. Minimizing what we defined as our performance measure can be achieved just because cars get stuck outside of the network or in a specific point of the network, not letting them arrive to most of the sections of the network, that stay with good speeds because they have smaller flows and not because there is a real improvement. So, at least two objectives must be defined: the best gain in terms of our performance measure, and maximum flow through the network.

Genetic algorithms concepts have been evolved to allow multi objective formulations.

Multi objective genetic algorithms are becoming popular and being applied with success to many network design and network optimization problems (Gen, Cheng, & Lin, 2008) and seems a good way to deal with our problem too.

Multi-objective genetic algorithm implementation

Deciding to take advantage of an already existing multi objective genetic algorithm overall structure that was flexible enough to customize it to our new problem, we have chosen MATLAB genetic algorithm and direct search toolbox (The Mathworks) for our multi objective genetic algorithm implementation.
Following, the parts that have been customized to our specific problem are presented in more detail.

Encoding

In a network with m turns to be taken in account in the decision of which turns must be open and closed, an individual, this is a topological network configuration, can be completely defined by a vector T of dimension m, with T_i taking the value 0 if the turn i is closed (turn prohibited) and value 1 if the turn is open (turn permitted).
In a similar way, considering that the network has n streams whose direction can be changed, being a stream a group of consecutive sections whose direction can only be changed together, a vector S of dimension n can be defined. For each individual to consider S_j would take value 0 if stream j has the same direction as in the normal network and would take value 1 if it has the opposite direction. This vector S can be constructed from vector T, being not necessary to define a different individual is useful in the application of the genetic operators.
Genetic operators as well as other steps from the algorithm require accessing descriptive information from the base network. The information required can be organized in 5 tables:

- Sections table – each line represents a stream j, columns are in this order: Id in Aimsun of the initial section of the stream in the original direction; Id in Aimsun of the ending section of the stream in the original direction; Id in Aimsun of the initial section of the stream in the opposite direction; Id in Aimsun of the ending section of the stream in the opposite direction; Id in Aimsun of the U-turn from stream in original direction to stream in the opposite direction; line/position of parallel streams if they exist.
- Turns table – each line represents a turn i, columns are in order: The line/position in Sections table of the origin section of the turn (if origin section way cannot be reverted section is not in sections table and this take value 0); The line/position in Sections table of the destination section of the turn (if origin section way cannot be reverted section is not in sections table and this take value 0); the turn ID in Aimsun; origin section Id in Aimsun; destination section Id in Aimsun; Turn Node Id in Aimsun; the class of the turn codified in 1 for U-turn, 2 for left turn, 3 for through turn and 4 for right turn.
- Actual Solution – The T vector for the individual that represent the actual solution, the normal network configuration.
- Connections – pairs of Ids from sections connected by fixed/unchangeable turns.
- OD pairs – first pair of numbers in each line are the Ids of the centroids of an origin/destination pair, the following pairs in the same line represent possible combinations for entrance/exit sections Ids for the same OD pair.

Initialization

Sometimes it is convenient that specific individuals make part of the initial population. When so, those individuals are specified and the other individuals of the initial population are created by the creation function. In this particular problem it is convenient that at least the individual representing the actual solution, this is the normal network configuration is defined as being part of the initial population. In order to complete the initial population a function named Creation and summarized in the scheme of Figure 2 was defined.

Fitness Evaluation

Fitness Evaluation is made by Fitness function, this function receives an individual to test as an input, translates the description of the individual to Aimsun language, calls Aimsun to run some replications for the period while the incident lasts with options discussed for situation 4 and return the evaluation given by Aimsun. Four objectives are considered:

- Minimize the difference between network performance measure for situation 4 and situation1.
- Minimize the difference between network flow for situation 4 and situation 1.
- Minimize the difference between lost vehicles for situation 4 and situation 1. If changes in the network get drivers in a part of the network from where they pass to be unable to reach their destination, they leave the network from a different exit point, being counted as lost by Aimsun. But in reality what happens is that the new path to reach destination is not completely inside the part of the network represented in Aimsun, and part of the individual effect of that driver on the network performance is under evaluated by Aimsun. This objective was introduced with the aim of avoiding too much distortion in the data that would turn to be incomparable to undistorted data.
- Minimize the number of nodes changed because of safety issues.

Most objectives are considered in terms of minimizing differences to normal situation because we see this work as an approach to deal with unexpected negative situations whose effects we wish to eliminate or at least mitigate and quickly return to the normal conditions to the network. Also with this formulation it is clear that the optimum to achieve for those objectives is the zero and is easier to evaluate if the algorithm is progressing well or still far from optimal solution.

The scheme in Figure 3 details the Fitness function.

The figure makes reference to two tables not yet mentioned:

- Actions Table – each column has a list of Aimsun Ids, by order of columns, from: turns closed; turns to close with incident; turns to open with incident; turns to open after change directions; turns to force; turns to close after change directions; sections to change.
- LstNodesTurns Table – each line represents a node and register de node Aimsun Id, if the node is going to be changed and the Aimsun Ids of the node turns to be open for some period of time.

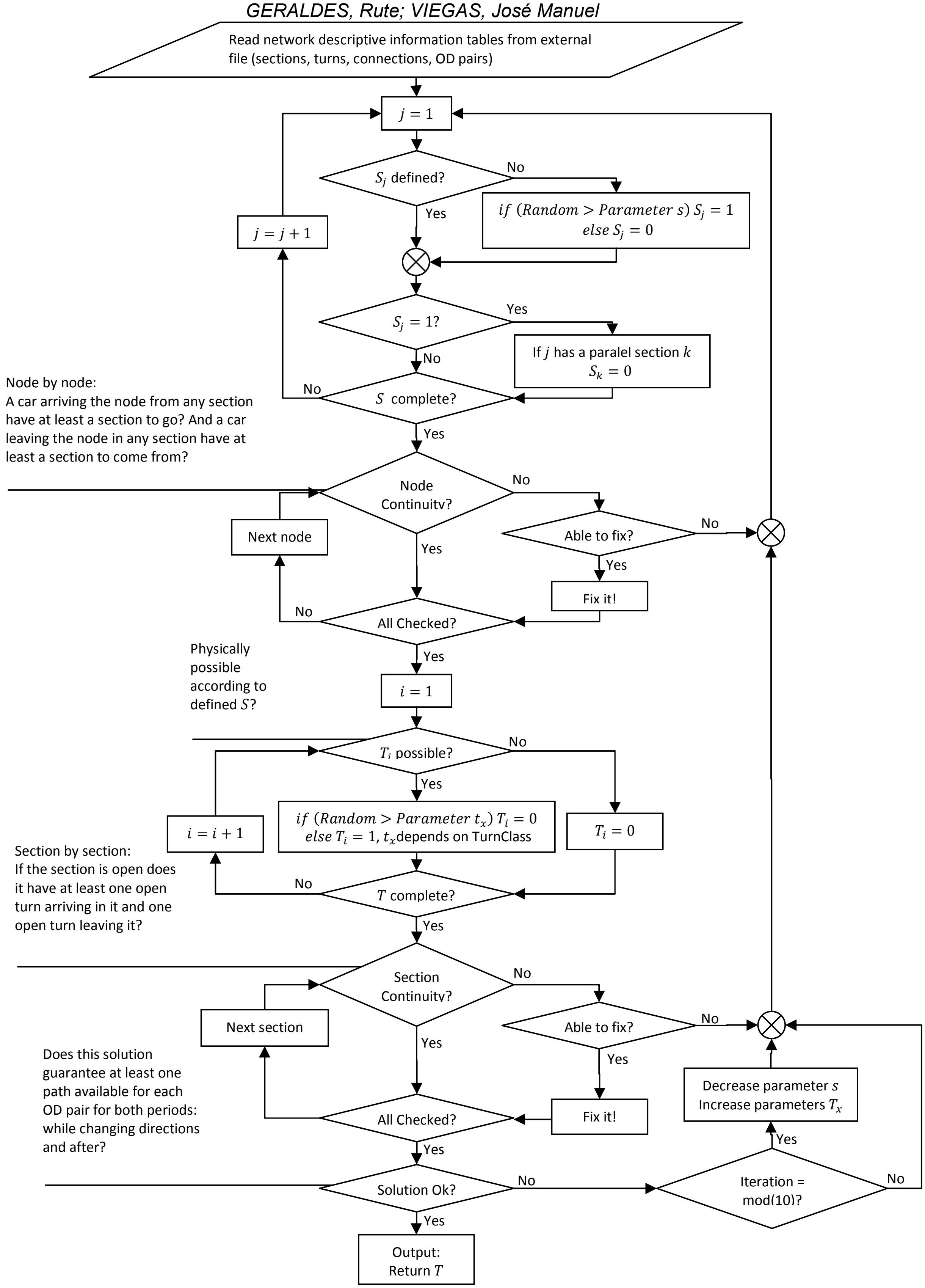

Figure 2 – Creation function

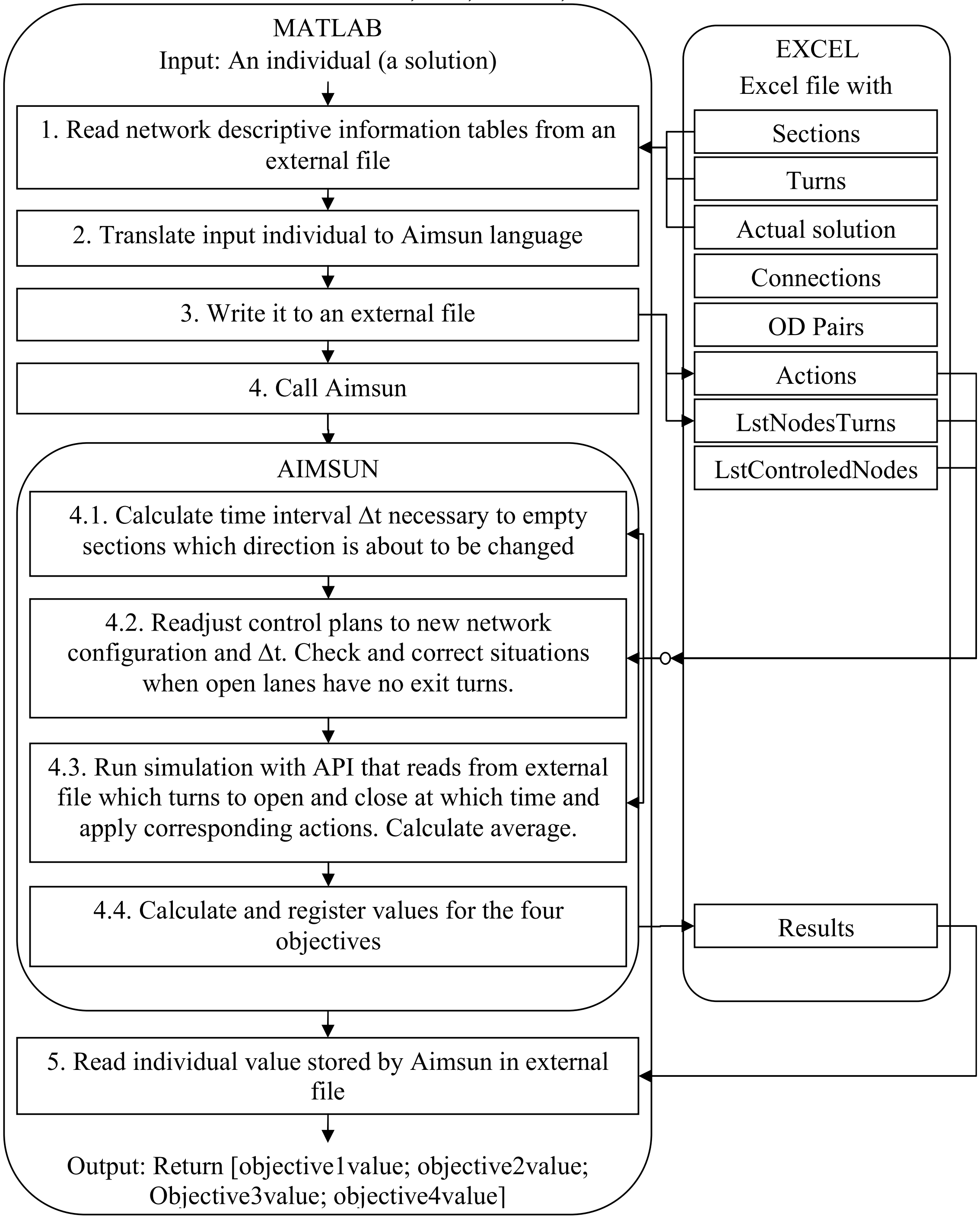

Figure 3 – Fitness function

Genetic Operators

Crossover

Crossover function receives a list of y parents do be considered to crossover and repeats the steps represented in Figure 4 in a schematic way y/2 times in order to return y/2 children.

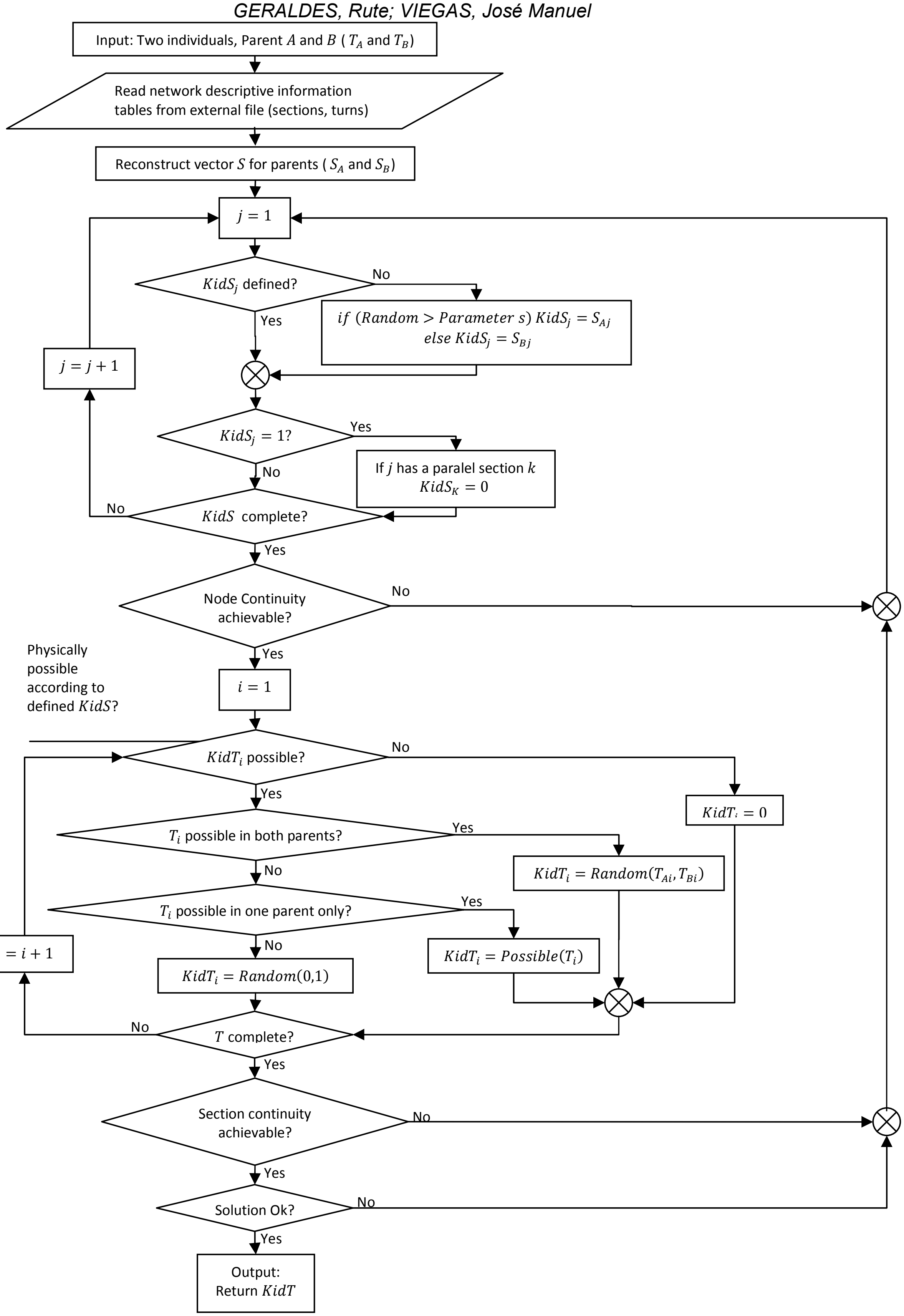

Figure 4 – Crossover function

Mutation

Mutation function receives a list of w parents to be considered for mutation and repeats the steps represented in Figure 5 in a schematic way, w times in order to return w mutated individuals.

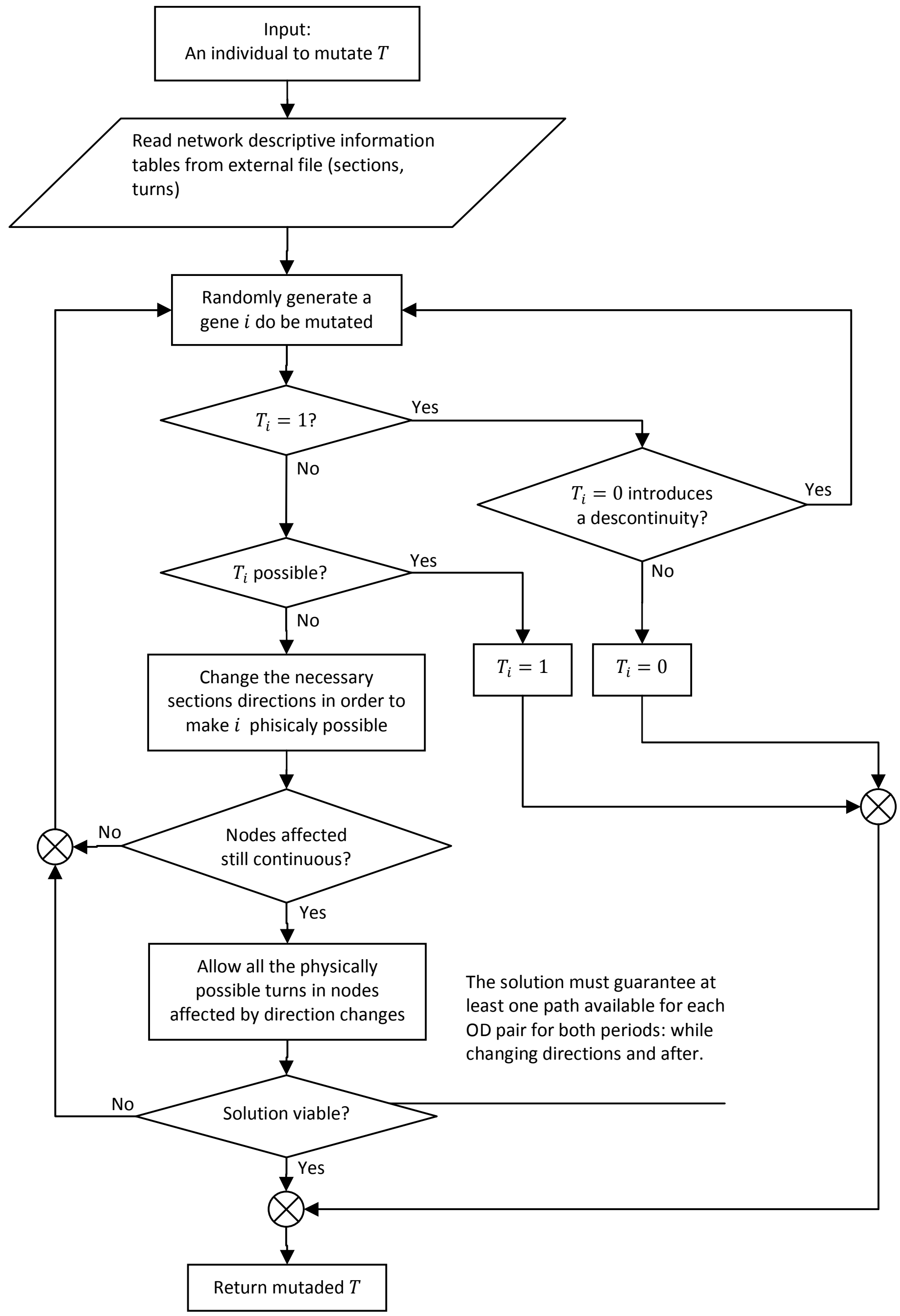

Figure 5 – Mutation function

Selection

For Selection the Matlab tournament selection was used.

Other problem settings

Regarding other problem settings like crossover fraction, migration, stopping criteria and multiobjective problem settings, we started with Matlab default parameters but tests are still running in order to tune them.

REFERENCES/ BIBLIOGRAPHY

Fitzpatrick, J. M., & Grefenstette, J. J. (1988). Genetic algorithms in noisy environment. *Machine Learning 3* , pp. 101-120.

Gen, M., & Cheng, R. (2000). *Genetic algorithms & engineering optimization.* John Wiley & Sons.

Gen, M., Cheng, R., & Lin, L. (2008). *Network models and optimization, multiobjective genetic algorithm approach.* Springer.

Geraldes, R., & Viegas, J. M. (2010). Real time changes of road network topology as an approach to mitigate the effects of incidents in urban congestion. *12th World Conference on Transport Research.* Lisbon.

Information Society Technology Programme. *Prime - Prediction of congestion and incidents in real time, for intelligent incident management and emergency traffic management; Dissemination and use plan report.* Information Society Technology Programme.

Kocks Consult GmbH. (1978). *Gesamtverkehrplan Grossraum Nurnberg.* Kurzfassung.

Ortúzar, J. d., & Willumsen, L. G. (2001). *Modelling transport, Third Edition.* John Wiley & sons, LTD.

Peeta, S., & Ziliaskopoulos, A. K. (2001). Foundation of dynamic traffic assignment: the past, the present and the future. *Networks and Spatial Economics, I* , pp. 233 - 265.

Sheffi, Y. (1985). *Urban Transportation Networks: Equilibrium analysis with mathematical programming methods.* Englewood Cliffs, N.J.: Prentice Hall, Inc.

The Mathworks. *Global Optimization Toolbox 3, user's guide.*

TSS - Transport Simulation systems. (April 2010). *Microsimulator and Mesosimulator Aimsun 6.1 User's Manual.*