

Large Scale Traffic Simulations

Kai Nagel^{1,2}, Marcus Rickert^{1,3}, and Christopher L. Barrett^{1,2},

¹ Los Alamos National Laboratory, TSA-DO/SA MS M997, Los Alamos NM 87545, U.S.A., kai@lanl.gov, barrett@tsasa.lanl.gov, rickert@tsasa.lanl.gov

² Santa Fe Institute, 1399 Hyde Park Rd, Santa Fe NM 87501, U.S.A.

³ Zentrum für Paralleles Rechnen ZPR, Universität zu Köln, 50923 Köln, Germany

Abstract. Large scale microscopic (i.e. vehicle-based) traffic simulations pose high demands on computational speed in at least two application areas: (i) real-time traffic forecasting, and (ii) long-term planning applications (where repeated “looping” between the microsimulation and the simulated planning of individual person’s behavior is necessary). As a rough number, a real-time simulation of an area such as Los Angeles (ca. 1 million travellers) will need a computational speed of much higher than 1 million “particle” (= vehicle) updates per second. This paper reviews how this problem is approached in different projects and how these approaches are dependent both on the specific questions and on the prospective user community. The approaches reach from highly parallel and vectorizable, single-bit implementations on parallel supercomputers for Statistical Physics questions, via more realistic implementations on coupled workstations, to more complicated driving dynamics implemented again on parallel supercomputers.

1 Introduction

Nobody likes traffic jams. Yet, they are only the most visible feature among a variety of related problems: Subways which fail to go where or when you need them; pollution of inner cities; etc. Many of these are results of poorly designed transportation systems. However, what is a good design? In our complex world, such a question is not easy to answer. Addition of new streets may *increase* congestion by concentrating formerly spread-out traffic onto one through route [1]; introduction of a transit system may *increase* pollution by making the car which previously was taken to work now available for short trips all with a cold engine [2]; a transportation infrastructure investment payed for by a certain group may actually turn out to benefit a completely different sub-population (winner/looser analysis); a new major arterial meant to relieve congestion may attract new developments along this new arterial, making congestion worse in the long run (induced demand).

There is more and more agreement between transportation professionals that a useful planning tool for such situations is a transportation microsimulation. In such a microsimulation, each traveler is represented as an individual object in the simulation. That makes it straightforward to separate out winners and losers; to “look” for vehicles with cold engines causing excessive pollution; etc. Yet,

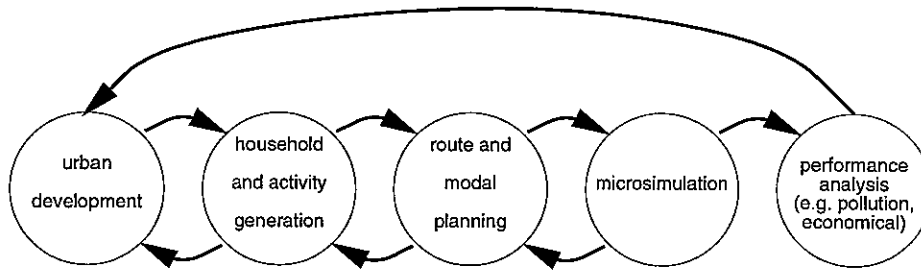


Fig. 1. The TRANSIMS design.

a problem is how do you “drive” such a microsimulation, i.e., how do travelers decide their next move at intersections or at transfer points (train stations etc.)?

The traditional answer to this questions has been “turn counts”, i.e. numbers at each intersection which tell you which percentage of vehicles makes left or right turns, respectively. It is fairly clear that this will not work as soon as the infrastructure changes affects people’s behavior; e.g. a left turn which has been used heavily before the change is now no longer used much, rendering the turn counts for this particular intersection irrelevant.

It is also fairly obvious that random selection (annealed randomness) often favored by (statistical) physicists has a good chance of not being very helpful in the very non-homogeneous structure of transportation systems.

2 TRANSIMS

It thus seems that the only answer is to drive the traveler objects in the simulation by something which emulates real-world behavior, i.e. by intentions. This, and its realization into a practical computer code, is the core of the TRANSIMS project [3, 4, 5, 6, 7] (see also [8]).

With intentions, one is very soon faced with a consistency problem. Real people presumably plan their trip before they leave their current location (meaning one needs simulated planning), but are open to deviate from the plans for example when the conditions they encounter are much different from what they expected (meaning one needs on-trip planning). Furthermore, the intentions somehow have to be generated in the computer in the first place.

The TRANSIMS approach to this problem is to parcel out the different parts of this process (see Fig. 1):

- Population and activities generation: Stochastically generate a population of individuals for a given geographic area such that the demographics of this generated population matches demographic data. Then, for each individual, generate activities such as work, shopping, social activities, which that individual wants to perform under some scheduling restrictions (e.g. go to work at 9 am; go shopping once a week; etc.).

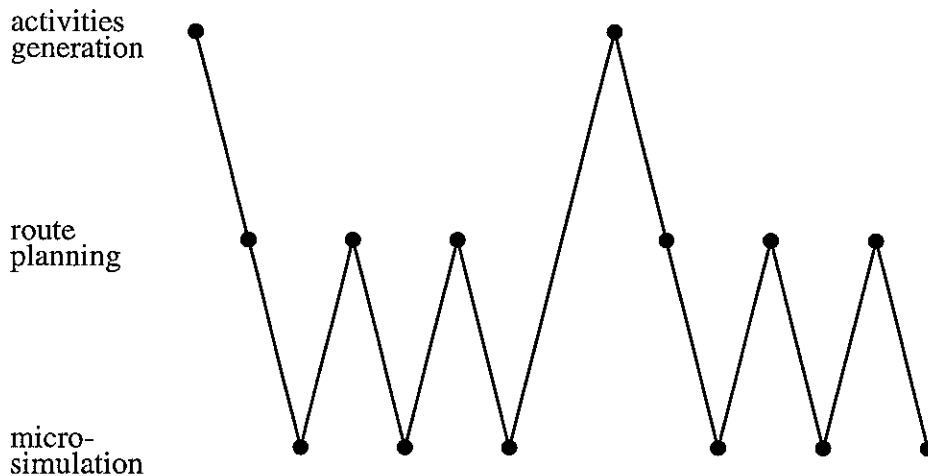


Fig. 2. Visualization of feedback cycles between activity generator, planner, and microsimulation.

- Trip chaining, modal choice and route planning: The activities are combined into trip chains, and the trip is routed on the transportation system, including modal choice
- Microsimulation: The trip plans are executed in a detailed microsimulation of the transportation system
- Analysis: The output of the traffic microsimulation is input into analysis modules, such as air quality, or measures of efficiency analysis

Despite the separation of the modules, it is clear that there are backward causalities. For example, if the microsimulation displays congestion, people will react by choosing different routes. If that does not help, they will re-schedule their activities. If nothing helps, they will maybe relocate to a more convenient location.

This backwards causality is what causes one of the computational challenges. In order to sort out the causal interdependencies, it is necessary to run the microsimulation which results in a new cost function (travel-time) on the infrastructure, then to re-plan according this new cost function, then to run the microsimulation again, etc., until some relaxation criterion is fulfilled. After that, one probably has to re-run the activities generation, adapting activities to what can actually be achieved in the given transportation system. This triggers in return a completely new relaxation cycle between route planner and microsimulation, etc. (see Fig. 2). In consequence, for certain problems the microsimulation may have to be run hundreds of times, thus demanding ultra-high computing speeds.