Traffic jams for the travelling salesman

Solving the stochastic time-dependent travelling salesman problem with a genetic algorithm

Abstract

The standard travelling salesman problem has fixed travelling times between cities. In the case of traffic jams, fixed travelling times can not be assumed: the travelling time is stochastic and depends on the time of departure. In this article stochastic, time-dependent travelling times are modeled with a lower and upper bound function and a profile, which gives the probability distribution between the lower and upper bound. Necessary and sufficient conditions are derived under which a later time of departure decreases the probability of arriving before a certain time both in terms of the lower and upper bound function (for an arbitrary profile) and in terms of the parameters describing the expected travelling time (as a function of the time of departure).

Solving the stochastic time-dependent travelling salesman problem with time frames is subject of the second part of the article. Expressions have been derived for the goal function, using the concept of robust solutions and a genetic algorithm has been developed to find good solutions. An Or-opt algorithm has been combined with a new crossover-operator. The crossover-operator reduced the calculation times, or resulted in better solutions with a slightly longer calculation time.

Modeling stochastic time-dependent travelling times

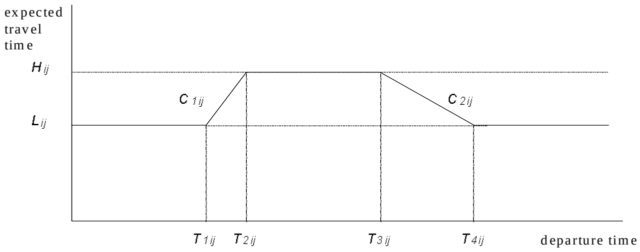
No model has been found in literature, which combines both aspects. The model in this article has been based upon a time-dependent model form Ahn and Shin [1] and a stochastic modeling concept from Kao [2]: the travelling times are described using probability functions.

In this model the travelling time $R(t)$ with a time of departure t is bounded by a lower bound function $l(t)$ and an upper bound function $u(t)$ with $0 < l(t) < u(t)$ for each t. The travelling time's distribution is now defined using a profile $G(.)$ independent of the time of departure t. This profile should be a probability distribution defined on [0,1]. The probability function for the travelling time is described by:

$$F_t(z) = \begin{cases} 0 & \text{als } z \le l(t) \\ G\left(\dfrac{z - l(t)}{u(t) - l(t)}\right) & \text{als } l(t) \le z \le u(t) \\ 1 & \text{als z} \ge u(t) \end{cases}$$

It can be proven that if travelling times follow this distribution, later departure times can result in earlier arrival times. Obviously, later departure times should be a "disadvantage". It can be proven that for an arbitrary profile defined on [0,1] with a non-zero probability for each value on this interval, the probability of arriving before a certain time decreases if and only if for every ($t_1, t2)$ with $t_1 < t_2$: $t_1 + l_1 < t_2 + l_2$ and $t_1 + u_1 < t_2 + u_2$ (the *'decreasing-probability-property'*)

The second property, the model should have, is that the expected value of the travelling time corresponds with the following function (i denotes the departure city, j the city of arrival)



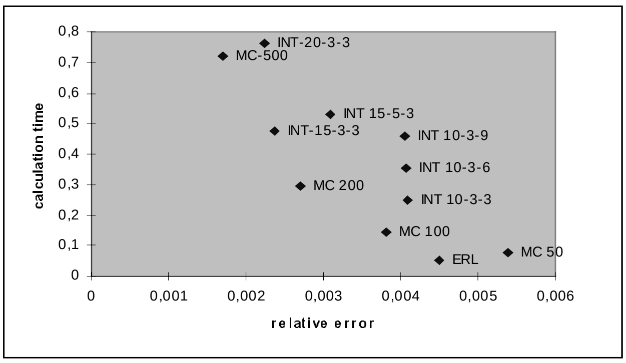
It can be proven that for a given probability function $G(.)$ defined on [0,1] with a non-zero probability for every value on this interval and an expected value EG, functions $l_{ij}(t)$ and $u_{ij}(t)$ exist, which have the *decreasing-probability-property* if and only if $c_{2,ij} > -1$

Calculating the objective function

In the first section a model is described for stochastic time-dependent travelling times. As the individual travelling times are stochastic, the total travelling time is stochastic as well. In this research, objective functions have been used, based upon the concept of robust solutions and taking into account time frames in the cities.

To estimate the objective value of a specific tour, the probability distribution of the arrival time in city i should be estimated. Three methods have been developed and tested: a Monte-Carlo method, a method based on Erlang approximation of the arrival time distribution function and a method based on integral approximation.

Several versions of the three methods have been tested on 15 randomly generated problems. In the graph below both the calculation time and the relative error are shown. It can be concluded that the strength of the Erlang method is its calculation time. If a more accurate value is needed, the Monte Carlo method seems more appropriate than the integral approximation method.



ERL = Erlang approximation
MC-N = Monte Carlo method with sample size N
INT A-B-C = Integral approximation in A points of the distribution function with B different step sizes and C steps

Solving the problem with a genetic algorithm

In this section, a heuristic based on genetic local search is discussed. Genetic local search is a general approach and certain operations have to be defined. These operations are applied to a family of solutions (instead of one solution as is the case in local search algorithms). The first operation is a reproduction-operator, which constructs a new family or generation by selecting members from the previous one. In general, the better a solution the large the probability that the solution survives. Secondly a crossover is applied; the crossover constructs new solutions ('children') from two old solutions ('parents'). These new solutions are improved using a local search algorithm. The procedure is then repeated until the pre-established number of generations has been reached.

The crossover-operator should be defined in a way that children resemble their parents, but not too closely. In this research a new crossover has been developed, which has some resemblance to the Order Crossover [3,4,5]. To construct a new tour from two parent tours the following steps are executed:

1. randomly select a path in one of the parents (both the starting point and the length are random)
2. change the sequence in this path according to the sequence in the other parent
3. repeat this procedure for the other parent

The following example illustrates this principle. In the case of six cities, the parents are [1,2,3,4,5,6] and [6,5,2,1,4,3,]. In the first parent, the path [3,4,5] is randomly selected. The first 'child' now is [1,2,5,4,3,6] as the cities 3,4,5 are visited in the sequence 5,4,3 in the second parent.

The Or-opt algorithm [6] has been used to improve the members of a generation.

This algorithm has been tested for four versions of the algorithm, in which the number of generations, the number of tours in the first generation and the survival rate for the worst solution in a generation varied. Notice that the first version is equal to an Or-opt algorithm, which is repeated 20 times. The algorithm has been used in combination with the Monte-Carlo method to solve 150 randomly generated problem instances with 10 cities. The results are summarized in the table below.

	1 – 20 – 0.3	3 – 8 – 0.3	3– 12– 0.3	5 – 8 – 0,6
number of generations	1	3	3	5
number of tours in first generation	20	8	12	8
survival rate worst tour in a generation	–	0.3	0.3	0.6
average rank	2.75	2.71	2.38	2.17
average calculation time	326.63	247.35	350.61	377.55

There are clear differences in the performance of the different versions, although in 30 of the 150 cases all four versions found the same solution. Combining the Or-opt algorithm with a genetic algorithm reduced the average calculation time considerably or resulted in significantly better solutions with a slightly longer calculation time (comparing 1 – 20 – 0.3 with 3 – 12 – 0.3 using a sign test at a significance level of 0.001).

Literature

1. Vehicle-routing with time windows and time-varying congestion, B-H. Ahn, J-Y. Shin, Journal of the Operations Research Society, 28, 1980, p. 1018-1021.
2. A preference order dynamic program for a stochastic travelling salesman problem, E.P.C. Kao, Operations Research, 26(6), 1978, p. 1033-1045.
3. Genetic algorithms for the traveling salesman problem, Jean-Yves Potvin, Annals of Operations Research 63(1996), p. 339 – 370.
4. A study of permutation crossover operators on the traveling salesman problem, I.M. Oliver, D.J. Smith, J.R.C. Holland, in Proc. 2nd Int. Conf. on Genetic Algorithms (ICGA ’87), Massachusetts Institute of Technology, Cambridge, MA (1987), p. 411 – 423.
5. Genetic Algorithms in Search, Optimization and Machine Learning, D.E. Goldberg, Addis Wesley, 1989.
6. The Traveling Salesman Problem: a guided tour of combinatorial optimization, E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, D.B. Shmoys, John Wiley & Sons, Chicester, 1985.