ARGESIM COMPARISONS

Restaurant Business Dynamics – Definition of a new ARGESIM Comparison - C16

M. Gyimesi, F. Breitenecker, Vienna University of Technology; mgyimesi@osiris.tuwien.ac.at

A. Borshchev, Techical University St.Petersburg / XJTek St. Petersburg; andrei@xjtek.com

This comparison addresses modelling, simulation and optimisation of a discrete dynamic system. The business under examination is the restaurant business, although you can substitute this domain by any other domain dealing with branch offices in an unchanging market with a fixed number of people.

General Description: We are going to investigate a rectangular area with a fixed number of people and a (dynamically changing) number of restaurants. People are randomly distributed over the area with uneven density and do not move. The restaurants initially present in the model are distributed evenly. People go to restaurants from time to time with random intervals. Every time a person goes to a restaurant, he leaves there a fixed amount of money. People only go to restaurants that are located within a certain distance range from their home, and choose randomly in case there are many of those.

A restaurant accumulates its revenue during a week and at the end of the week summarizes the financial results and applies the running policies. A fixed running cost is subtracted from the gross weekly revenue, and then the tax (fixed percent) is applied. The remaining profit is analyzed. If the profit is greater than some fixed threshold value, the restaurant with some probability launches a new one at the location with the best ratio of people to restaurants density. Otherwise the restaurant closes down, again with some probability.

Multiple simulation runs will be used to identify the asymptotic behaviour of the system. In addition you should try to optimise the income of the government (accumulated tax payment) respectively the anticipated income of a restaurant.

There are no restrictions how to build the model and run simulations. You are invited to use high level simulators of different kind (object oriented, process oriented, agent based …) as well as low level coding and any mixture of it.

People: 3000 people live in a rectangular area with *Width* 600 and *Height* 400 (in a larger scale each person can be seen as aggregation of e.g. 1000 people). Every person belongs to one of the five cities, (see Fig. 1).

The coordinates of the city centers and the percentage of the whole population living in a city are given in Table 1. Person's location with respect to the city center is defined in polar coordinates (*angle*, *radius*) where angle is uniformly distributed, and radius is distributed triangularly from 0 to *MaxR* with mean at 0. If a randomly chosen location of a person is outside the area, a new location is calculated.

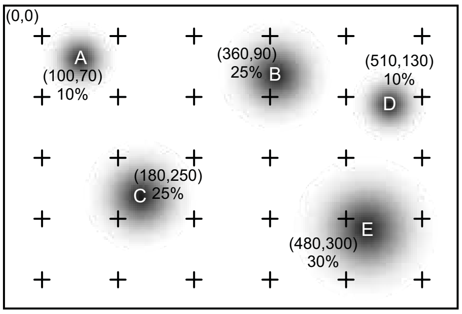

Fig. 1: Area with cities

	X	Y	MaxR	% of People
City A	100	70	100	10
City B	360	90	250	25
City C	180	250	250	25
City D	510	130	100	10
City E	480	300	300	30

Table 1: Location and largeness of the cities

People go to restaurants from time to time with intervals in between distributed discrete uniformly from 1 to *Maximum Dining Interval*. The restaurant to go to is chosen randomly from those located within *Range* from the person's home and with equal probabilities, no history is taken into account. A visit to a restaurant takes zero time and results in leaving there a *Dinner Cost* – flat for all restaurants.

Restaurants: Initially there are 30 restaurants evenly distributed (arranged) across the area in five rows, six restaurants in each; the horizontal distance between restaurants is 100, and the vertical is 80 (see Fig. 1). Restaurants are open every day. Do not model restaurant capacity and assume a restaurant can room any number of people.

The only source of revenue for a restaurant is what people pay when they visit it, so the weekly revenue is proportional to the number of visits during a week with coefficient Dinner Cost. At the end of each week a restaurant has to pay the weekly Running Cost (fixed) and the weekly Tax to the government – the fixed percent of what remains, zero if the revenue is smaller than the Running Cost. Whatever money remains after running cost and tax deduction is called Profit (can be negative).

When weekly profit is calculated, two rules (policies) are applied:

(a) if *Profit* is higher than fixed *Profit Threshold*, open a new restaurant at the best possible location (see below) with *Opening Probability*;

(b) if *Profit* is lower than *Profit Threshold*, close down with *Closing Probability*.

Finding location for a new restaurant: The best location to launch a new restaurant is found by partitioning the whole area into square cells of size 20x20 (there are 30 * 20 = 600 such cells) and calculating the ratio of *People Density* / *Restaurant Desity* for each cell. The *People Density* is simply the number of people living in the cell, whereas the *Restaurant Desity* is calculated as

$$\sum_i 1/D^k(cell\_center, rest_i)$$

While *D* is the distance between the cell center and the restaurant *i, k* is a weighting coefficient choosen appropriately (and investigated in one of the tasks). The cell with the maximum ratio of the densities is chosen and the new restaurant is placed randomly and uniformly within that cell.

Parameters: You are invited to experiment with all of the area settings, parameters and distributions but to compare different solutions of this problem you are asked to use the following parameter values:

parameter	value
Total Number of People	3000
Initial Number of Restaurants	30
Area Width and Height	600 x 400
Maximum Dining Interval	8
Dinner Cost	1
Range	100
Running Cost	150
Initial Tax Rate	20%
Profit Threshold	350
Opening Probability	10%
Closing Probability	20%
k in Restaurant Density	4

Table2: Model Parameters

Calculating results: Depending on the tasks, the focus is on the accumulated tax – income for the government respectively the capital of the restaurants.

Model approach: Give a short explanation of your model approach (process oriented, agent based, event approach, activity scanning approach, directly programmed, etc.) and the used simulation environment, development environment or software environment, resp.

Task a – Time Domain Analysis: Build the model and simulate the system for 1, 5 and 10 years and show the total number of restaurants over time. Perform 50 simulation runs and calculate the average limit value of number of restaurants after the 5th year.

Task b – Tax Income Maximisation. Maximise the tax income for the government and show the dependence of tax income from *Tax Rate*. The higher the tax rate is, the more tax will be paid by a single restaurant, but otherwise fewer restaurants will survive – and vice versa. Start with a parameter variation of the tax rate. There should be at least one maximum. Analyse and discuss, if this maximum reflects the best possible solution in view of the government. You can use other optimisation methods or arbitrary control strategy.

Task c – Restaurants' Revenue Analysis. Analyse the expected revenue of new restaurants depending on the strategy for opening new restaurants.. As the income of a restaurant depends only on the place where it will be located with respect to population and other restaurants, the evaluation of the existing situation is the important factor to control restaurants revenue. Therefore, vary the parameter *k* in the function for Restaurant Desity from 0 to 6 in steps of 0.5 and indicate the best value for k (max. revenue).

Comments and further information: This comparison is intended to be a very general one. On the one side, variable structures are needed, as found in object-oriented approaches. On the other side, time advance is relatively simple, so that event-scheduling mechanisms are not a must, and consequently any programming system can be used (time advance is given by the discrete distributed Dining Interval - 1 to MaximumDining Interval days -, and after 7 days profit is calculated).

Of course, there are well suited approaches for this comparison, e.g. agent-based modelling – with dynamically changing spatially structures, etc. But also very classical programming approaches, more or less directly programmed, can be used, with high efficiency (compared to simulation systems). Furthermore, also implementations with activity scanning may be advantageous; activity scanning is usually listed as third world view in discrete event simulation – but never used (the others views are event scheduling and process interaction). Or one may make use of a statistical environment, where time advance is programmed in the interpreter of the system, or one may use a computer algebra system, etc.

If questions or remarks coming up during modelling, simulation or interpretation of this comparison, feel free to contact the authors or have a look at www.argesim.org, where you can find extended information.