**Appendix: Information on Web**

This book is a scientific report of a very solid piece of research. The research is mainly focused on [Scheduling of Automated Guided Vehicles](http://privatewww.essex.ac.uk/~hrashi/Overview%20of%20this%20research.htm) [[1]](#footnote-2)(AGV) in container ports. The problem is to schedule several AGVs in a port to carry many containers from the quayside to yardside or vice versa. This problem is formulated as a Minimum Cost Flow problem and then solved by Network Simplex Algorithm (NSA), Network Simplex plus Algorithm (NSA+), Dynamic Network simplex Algorithm (DNSA) and Dynamic Network simplex Plus Algorithm (DNSA+). The research contributions are NSA+, DNSA and DNSA+. NSA+ is faster than NSA. DNSA and DNSA+ repair the previous solution when any changes happen.

To test the model and performance of the algorithms in our implementation, many jobs have been generated. Their sources, destinations and the distance between every two points in the port have been chosen randomly. As it can be seen in Figure Appendix-1, our software, which has been implemented by C++, running on a WinTel PC, can find the global optimal solution for three thousand jobs within two minutes.



Figure Appendix-1: CPU-Time required solving the graph model

**Overview of this research**

This research concerns itself with the scheduling of Autonomous Guided Vehicles (AGV’s) in a port. Port components that are relevant to our problem include berths, Quay Cranes (QC), container storage areas, and a road network. A transportation requirement in a port is described by a set of jobs. Each job is described by:

(a) The source location of a container.

(b) The target location, where the container is to be delivered to.

(c) The time at which it is available for pickup or dropoff on the quayside.

Any delay will incur heavy penalties. Given a number of AGVs and their availability, the task is to schedule the AGVs to meet the transportation requirements.

**Assumptions**

* The layout of a port container terminal is given. Also it is assumed the vehicles move with an average speed so that there are no **Collisions**, **Congestion, Livelock** or **Deadlock** problems**.**
* The travel time between every combination of pickup /dropoff points is provided according to our layout.
* Every AGV can transport only one container. Also it is assumed that the start location of each AGV at the beginning of the process is given.
* Rubber Tyred Gantry Cranes or Yard Crane resources are always available, i.e. the AGVs will not suffer delays in the storage yard location or waiting for the Yard Cranes.
* The source and destination of container jobs over the port are given.
* For each QC, there is a predetermined crane job sequence, consisting of loading jobs, or unloading/discharging jobs, or a combination of both. For each loading/discharging job, there is a predetermined pickup/dropoff point in the yard, which is the origin/destination of the job.
* Appointment time of every container job at its source/destination on the quayside is given.
* For the dynamic aspect of the problem, it is assumed that the number of vehicles is fixed, but the number of jobs and the distance between every two points in the port may be changed.

**Development**

Our software consists of the optimisation, scheduling and a simulation program. The software can find the global optimal solution for three thousand jobs and ten million arcs in the graph model within two minutes by a WinTel PC. Figure Appendix-2 shows the main form of the software.

Some important features of our program are described briefly in the following sections:

* The user can define a few ports, the number of blocks in the yard, the number of working positions or crane and the number of Automated Guided Vehicles in each port.
* A facility to generate the distance between different points in the yard or in the berth has been considered. At the first step, this distance is generated randomly, but it is modifiable by the user.
* For static and dynamic fashion, a few container jobs might be generated, which have to be transported from their sources to their destinations. Either the source or the destination of them is the quayside, which is chosen randomly by the Job-Generator. There are three options for Quay Cranes: single crane and multiple cranes randomly and circular. In the first option, crane number 1 is selected to handle the job whereas in the second option one crane, among several cranes in the berth, is determined to handle the job. In the latter option, choosing the crane number is circular; the first job for the first crane, second job for the second crane and so on. After the next job is assigned to the last crane, the turn goes to the first crane.



Figure Appendix-2: The main form of the software

* At the start of the process, the start location of each vehicle may be any point in the port. The user can define or change the ready time of the vehicles at the start location and the location as well. But at the first stage, we generate them randomly.
* The initial time for the operation, the time window of the cranes and vehicles should be defined by the user. The first parameter plays a role as the ship-arrival time; the second one means how long it takes for every job to be picked up or dropped off by the crane. The last one is the time for the vehicle to pickup/dropoff a job from/to the crane. We assume some default values for these parameters.
* The user can monitor some indices to measure the efficiency of the terminal. The waiting or delay time for every job, the number of jobs and the travelling and waiting times for every vehicle are calculated in the static and dynamic fashion.

**Some interfaces of our Software**

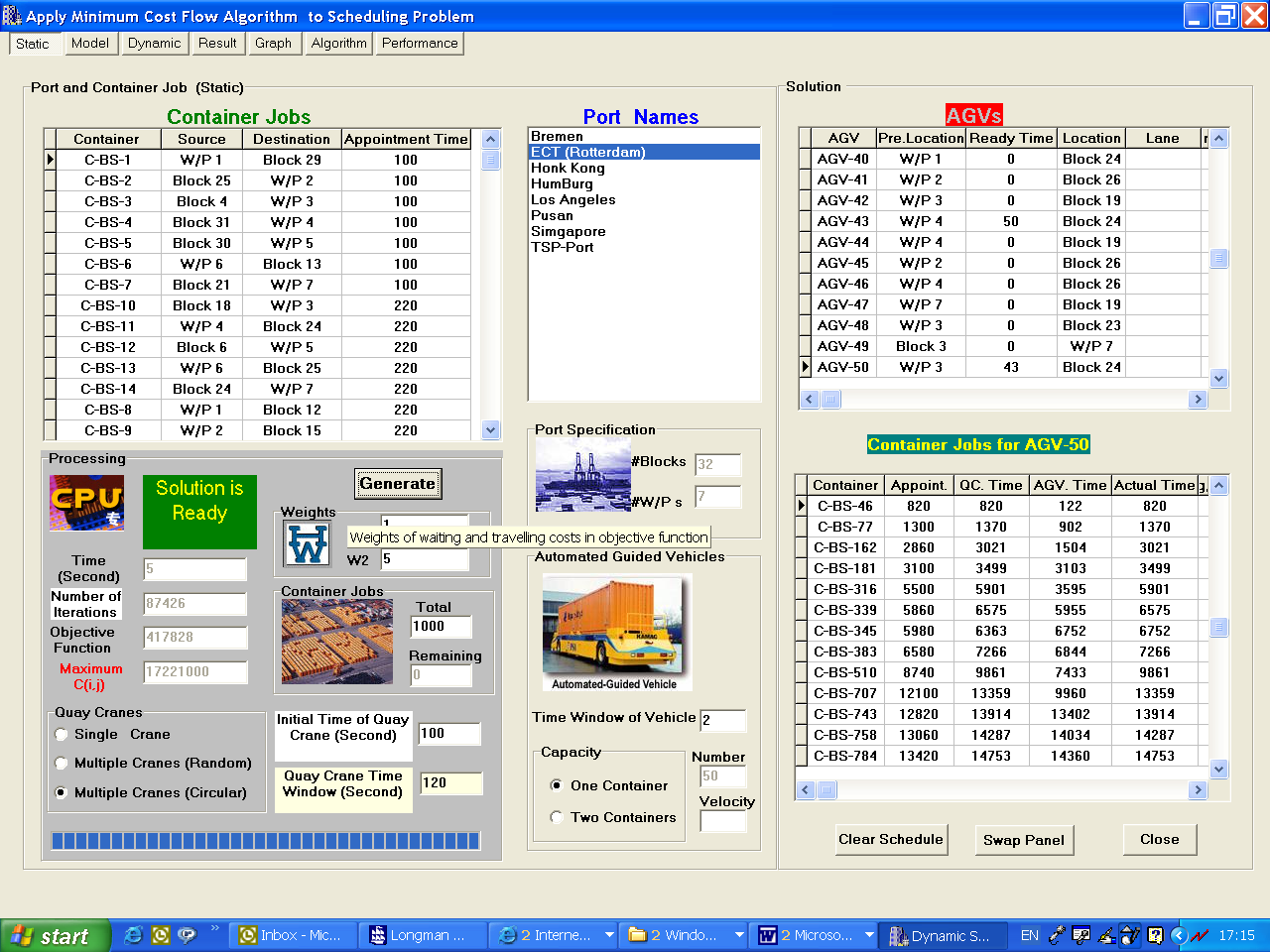


Figure Appendix-3: The output of Static fashion.

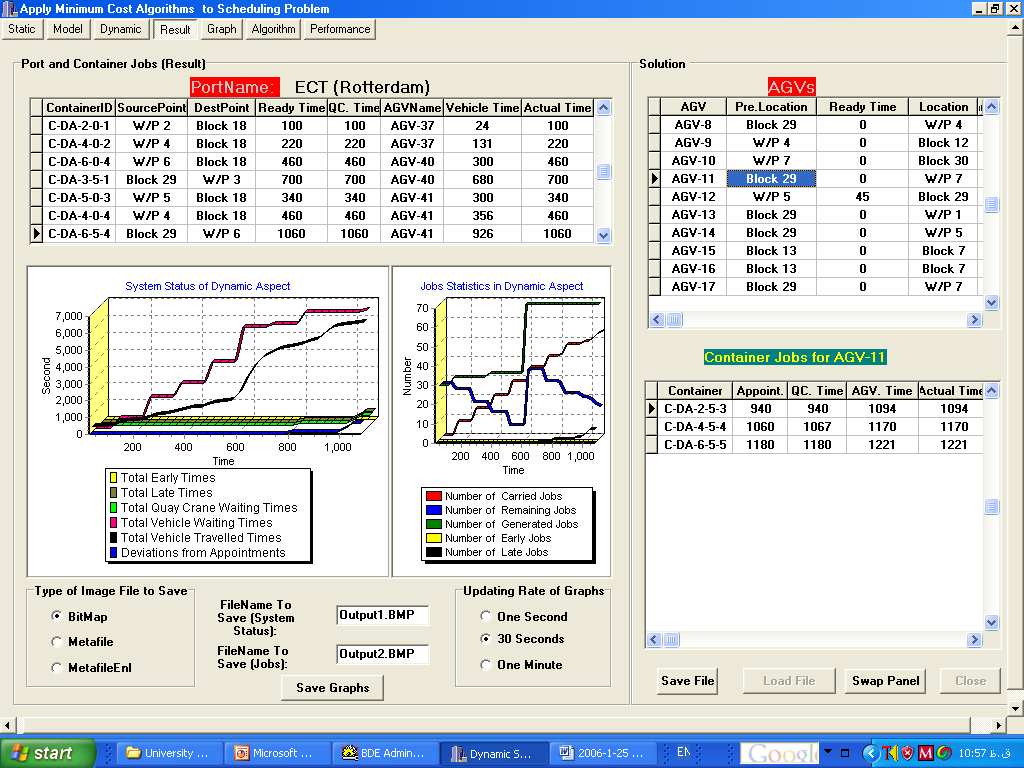


Figure Appendix-4: Monitoring some indicators of the output in Dynamic aspect

1. <http://www.bracil.net/CSP/autoport/Book.html> [↑](#footnote-ref-2)