cognito

White Paper

The Principles of Computer-Assisted Scheduling

COGNITO MOBILE WORKFORCE MANAGEMENT

Synopsis

In this paper we will cover the following topics:-

1. The principles of scheduling
2. Limitations of human schedulers
3. How computational schedulers help
4. The difference between static & dynamic scheduling
5. Application in practice
6. How Cognito can help you make the most of scheduling

1. The principles of scheduling

The efficient use of resources is a challenge for every company offering services that are provided by mobile workers. There are a range of issues, including meeting Service Level Agreements, maximising productivity, minimising the carbon footprint of a vehicle fleet and the equitable treatment of staff. Key to success in all these areas is the success that a company has in the scheduling of work, and the management of assets (including people, vehicles and parts). Many companies are now looking to computer-based scheduling systems to improve their scheduling performance, but as with any new technology there are a number of factors to be considered in both the selection of a product, and its integration into existing business process and IT systems.

This white paper will explain the fundamentals of a computer-based scheduling system. More details on the issues with regards to practical implementation and the benefits that can be realised by integrating the solution with other Mobile and Enterprise Systems are covered in the companion document 'Implementing successful computer-assisted scheduling'.

A good place to begin a discussion of the challenge of scheduling is the Travelling Salesman Problem (TSP). A salesman sets out on a series of appointments. The distance between all of these appointments is known, what is the shortest distance that he can travel and visit each appointment only once? This is a very simple scheduling problem, and it clearly has an exact answer, yet solving it by brute force, i.e. trying every possible route becomes rapidly harder as the number of points to be visited increases. This is because the number of possible solutions (solution space) increases much faster than the number of additional points. For example for 5 points there are 12 solutions, but for 10 there are 181,440. The increase in solution space with tasks and resources is "superexponential", i.e. faster than doubling at every step.

Mathematicians have tackled this type of problem by trying to come up with models that make approximations to the correct answer. They do this by taking an educated guess at its likely shape of the answer, and by doing so reducing the size of the solution space to be searched. The mathematical models then explore the reduced number of possible solutions in an informed manner.

2. Limitations of human error

The ability of machines to surpass human performance is a subject of continual interest and research. One surprising result is that humans are very good at reaching a reasonable result to a complicated TSP, and that the time taken is in fact proportional to the number of points[1] . This would indicate that the combination of human spatial perception and the way the brain works function as well as some of the most sophisticated mathematical approaches to this problem that have yet been devised. This would explain why a good human chess player, relying on experience and visualisation can match the performance of some of the biggest supercomputers in the world, and why researchers at the University of Washington are relying on human game players to solve protein folding problems as an improvement on a previous system using the brute force efforts of a peer to peer approach[2] .

1. "Human Performance on visually presented Traveling Salesperson Problems with Varying Numbers of Nodes" The Journal of Problem Solving, Vol1 No 1 (Fall 2006) pp20-32

2. Computer Game's High Score Could Earn the Nobel Prize in Medicine http://www.newswise.com/articles/view/540565/

The complexity of a TSP can be increased by adding weighting factors to the routes that the salesman may take, for example some roads may be toll roads and have a monetary cost, whereas others may be subject to traffic jams and have a time cost. As long as it is possible for these extra factors to be represented pictorially, or held in the mind of a human planner, it is likely that a reasonable schedule will be delivered.

The challenge comes when the number of factors that contribute to a solution move beyond simple weights that can be applied to a route. These would be familiar to any service organisation, and would include such factors as the customer SLA, the skill level of the engineer, the anticipated length of time for a job, the availability and location of stock, working hours, health and safety and access times to the customer premises. Other factors would be flexible rules (known as soft rules), for example an engineer who only works in a given geography, but who can be called on if a customer with a particularly tight SLA calls in with a problem.

In a typical company, the only way to solve a problem of this complexity is to split the engineers into teams with a typical engineer to planner ratio of 10:1. This is clearly expensive, both because of the number of planners required, as well as the inefficiencies caused by restricting the engineers to small groups, especially for teams that cover a large geographic area. In addition general assumptions, such as standard job times, further compromises the efficiency of the scheduling.

3. How computational schedulers help

At this point is becomes clear that some form of computational analysis is going to assist in the delivery of the most efficient and cost-effective schedule. Software schedulers take the idea of the TSP as a problem in two dimensional space and map the other hard and soft rules into a series of planes in n-dimensional space, in effect creating multiple TSP problems where a choice in one problem space influences the route costs in all the others.

The exact approach to approximating a solution to the scheduling problem is based on the mathematics of complexity theory. There are a number of approaches with names such as “simulated annealing” and “ant trail”. Taking simulated annealing as an example, the system estimates a solution and then changes certain variables to see if a better result is achieved (Figure 1). Over time, the amounts that the variables are changed is reduced. This is similar to the way a metal is gradually cooled (annealed) in order to maximise the size of the crystals that form, and therefore the strength. The blades of a turbine in a jet engine, for example, are produced from single metal crystals that are the result of an extremely careful annealing process. This may be visually represented as a series of undulations in a terrain, the idea being to jump over small peaks to look for deeper valleys between them representing lower energy states, and therefore better solutions.

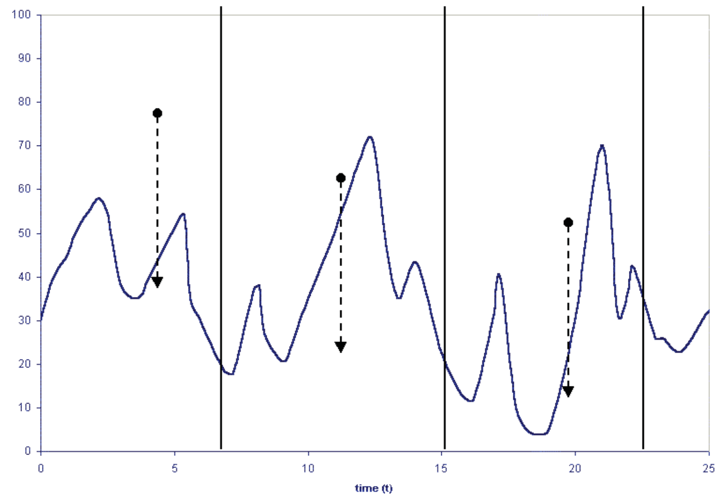

Figure 1. Gradual improvement in schedule using a simulated annealing approach

As you can imagine, this creates an opportunity for enormous amounts of computing horsepower, and so the elegance of the mathematics used to intelligently cut down on the searched solution space, as well as the efficiency of the algorithm used to search the remaining space will have a major bearing, not only on the closeness of the approximated solution to the perfect one, but on the time it takes to get there.

In fact there will be a law of diminishing returns, where allowing the scheduler to run on produces only minor improvements. Schedulers will typically give a score or confidence level with their output. It is important to note that this is not a guess as to the closeness to the ideal result, but merely states the rate at which the results were improving when the iterative process was suspended in order to deliver the schedule. If the confidence is low, the operator may chose to have the scheduler to continue to run in order to give a better result. If confidence is continually low, then it is likely that either a mistake has been made in creating the scheduling rules, or the rules themselves are too complicated for the scheduler to resolve and they must be simplified, or the scheduler replaced.

4. The difference between static & dynamic scheduling

There are two basic forms of scheduling: Static ("predictive") and Dynamic ("episodic" or "online"). Within each type, there are different methods of calculating schedules. Which type, and which method is most appropriate will depend on a number of key variables that are unique to any particular business. Selecting the most cost-effective scheduling technique is a skill that requires a broad understanding of your business, plus the underlying theory of scheduling, as well as an agnostic approach to the various commercially available solutions, and finally experience and expertise in the delivery and support of the system.

All schedulers work most efficiently if no events are fixed in time. Whenever an event such as an appointment is locked in place, the optimal solution is likely to be compromised. As the number of fixed points is increased, the scheduler becomes less and less effective. It is very important therefore that manual intervention in the schedule is restricted as much as is possible within the overriding needs of the business, and if possible fixed appointments are given a "time window" as opposed to a rigid and fixed time.

Static scheduling is the simplest type of scheduling. This is the case for businesses that carry out fixed work, for example courier delivery or planned maintenance. At the start of the process, all the assets, destinations and SLAs are known, and a calculation is performed to find the best possible solution. Once the solution is computed, it is translated into shift schedules and asset plans that are passed on to the field staff (Figure 2). Once in place the schedule does not change, and so little or no further optimisation is anticipated. Of course in practice there may be changes required, for example one engineer may go off sick, or get stuck in traffic, but this is the exception rather than the rule.

In dynamic scheduling, creation and execution are intertwined. Not all tasks are known before the start and so the schedule must adjust in real-time to new imperatives, such as new breakdown calls from customers with SLAs. The speed of delivery of an updated schedule will be of higher importance than the extra time taken to resolve the absolute "best-case" new schedule. In addition, scheduling jobs will be "drip fed" to field service personnel to make the entire system more flexible and therefore more reactive to customer needs. It is not practical to only send one job at a time to engineers, and it may be confusing to issue jobs and then rescind them, so an effective compromise is usually the next job plus a set number of subsequent ones.

The disruption caused by new tasks will be minimised if it represents a small change to the overall system. Therefore, a scheduler that can make its calculations on the whole universe of tasks and resources as opposed to one that relies on breaking the solution down into discrete parts (for example engineer territories) will provide lower costs overall.

Dynamic scheduling works by attempting to look at the most likely best solutions, as opposed to testing all that are possible. Reducing a large number of potential solutions to a smaller and manageable number of likely solutions brings down the time and computational cost involved in reaching a fair result. The computational difficultly depends on the number of degrees of freedom in the mathematical model, represented by the variables of business life, and this will be reflected in the number of hard and soft business rules that a scheduling product will allow the user to define.

Some dynamic schedulers produce faster results by estimating the subset of variables that are most likely to be impacted by the new information. This has the advantage of producing a faster result, with less computational horsepower required, but of course over time there will be a significant divergence from the results delivered by a dynamic scheduler that works on all the variables every time a change is requested. The lower cost of computing and evolution of scheduling engine algorithm makes dynamic scheduling increasingly cost effective.

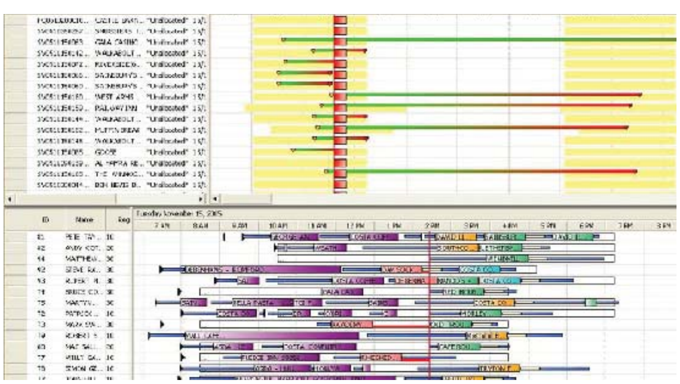

Figure 2. Typical Graphical representation of a schedule

5. Application in practice

The strict application of "hard" or absolute rules may yield higher-cost results than would be the case if certain rules are deemed "soft" and can be relaxed in certain situations for an additional cost. For example, an edict to eliminate overtime may be waived in some circumstances, whereas the requirement for drivers to take a break, being a statutory requirement and the subject of Health and Safety legislation may not. Some rules may be conditional, for example a painter may not be scheduled until after a plasterer, plus a suitable interval of time from the completion of the job to allow the plaster to dry. Some jobs may take longer than a day, so a scheduler must allow for tasks to remain uncompleted for an extended period.

It should be clear at this point that there are several key factors that affect the accuracy of a software scheduler, be it static or dynamic. In *decreasing* order of importance, they are:

1) The complete capture of all the hard and soft business rules and characteristics that can be factors in making a scheduling decision
2) The translation of these rules into the "route costs" that are used by the algorithms of the scheduling software
3) The ability of the scheduling software to scale to accommodate the number and type of rules captured in 1
4) The efficiency of the scheduling software's algorithm in arriving at an approximate solution, and its ability to recalculate based on all available data
5) The computing horsepower and time required to arrive at a solution

6. How Cognito can help you make the most of scheduling

Cognito has nearly 20 years experience in developing Mobile Workforce Management solutions for customers in a wide range of Facilities and Field Service Management industries. This gives us a detailed understanding of the business processes used by our customers

We have carefully evaluated different scheduling systems that are currently available on the market, allowing us to recommend the most suitable scheduling engine to a particular business Our unique skill in capturing a business process can be accurately translated into the mathematical rules that the scheduler will use, and our end to end Mobile Workforce Management expertise ensures that the scheduler is integrated fully with the full range of mobile systems and data in the business to deliver the best possible result.

About Cognito

With over 150 successful solutions in place and more than 19,000 subscribers, Cognito is the leading expert in Mobile Workforce Management systems and practice.

Cognito is the only company with the experience to tie together the key components of tracking, real-time data and scheduling and to deliver them to organisations as a powerful mixture of consultancy, design, delivery and support. Our Solutions impact directly on business performance by improving efficiency, reducing costs and increasing profitability.

Cognito Ltd
Benham Valence
Newbury
Berkshire
RG20 8LU
UK

www.cognitomobile.com
Tel: +44 (0)1635 508200
Fax: +44 (0)1635 550783