ILOG Optimization Suite International Users Meeting
Wednesday, July 9 1997

# CREW SCHEDULING SYSTEM

Prepared by: Accede Computer Solutions Pte Ltd

Benny Shum Kai Shun
Accede Computer Solutions Pte Ltd
Blk 104 Boon Keng Road, #03-06,
Kallang Basin Industrial Estate,
Singapore 339775
Tel: (65) 2988311
Fax: (65) 2985633
Email: accede@pacific.net.sg

**Introduction:**

The objective of this application is to automate the crew scheduling process of McDonald's Restaurants in Singapore and provide information for its various departments to analyse staffing and training needs. The application will be used to handle the rostering process for about 80 crews per store (restaurant). The application is developed using:

Ilog Solver and Ilog Planner for the optimisation engine
Ilog Views and Ilog InForm for GUI with Gantt chart(s)
Microsoft Access 95 as the database

The expected development completion date of this project is end of August 1997.

**Scope of the application:**

1. Create weekly schedule for the crew based on sales projections, crew availability and other parameters and constraints priority defined by the user. The scheduling engine is developed using Ilog Solver and Ilog Planner.

2. User will has the flexibility to edit the proposed schedule to handle exceptional scenario. The schedule will be displayed on a Gantt Chart and allow the user to manually drag and drop to make changes to the Gantt Chart. Ilog Views and Ilog InForm are used for doing this.

3. ll the necessary screens and forms which support the scheduling process are created using Ilog Views and Ilog InForm.

4. Identify training and staffing needs for each store.

5. Assist user in projecting labour cost.

6. Minimise data entry as most of the data such as sales projection and crew information will be downloaded from the host system.

7. Keep track of crew's training record, job grading and rotation requirement.

8. Upload the crew training information back to the Central Computer system for market wide analysis.

9. To provide customised reports for the management to monitor crew scheduling and training needs.

Running with extremely user friendly GUI Windows 95 features to minimise future conversion, upgrade and training costs.

**Glossary:**

Time interval: The unit of time that is used to generate the schedule and also what the sales projection is based on. The time interval can be in one hour or half-hour.

Sales Projection: The projected sales volume that will derive the number of crew required for the time interval.

Crew Positioning Guide (CPG): The chart that provides the number of crew required based on the sales volume for each hour.

Crew Availability Chart: The chart that provides the time period that the crew is available for work for each day.

Crew Station Rating: The rating that the crew acquired during the training for each station. The rating basically classify in three levels, i.e. Excellent, Good and Need Improvement.

**The Model:**

The Objective of the application is to assign a crew to meet the demand for each position. (e.g. 3 people are required for French Fries from 9:00 till 9:30) respecting a set of constraints such as crew availability (see below). The application will generate the weekly schedule for a store on a stand alone PC.

Due to the expectation of the user, the performance of the roster generation played a vital part when we designed the model. The application should not run more than 30 minutes for a weekly schedule running on a Pentium 133 with 16Mb of memory.

As a result, the model was broken into two phases in order to restrict the number of constrained variables as well as constraints.

Size of the problem:

Number of Stations: from 15 to 25
Number of People: from 30 to 80
Number of time intervals: from 30 to 40 in half hour intervals

**Phase 1:**

First phase of the application decides the number of crews that need to work for that day (based on the Sales Projection). The decision variables for this phase was expressed as booleans:

X[p][h][d] : where p is a given person, h is the starting hour and d is the duration that the crew need to work for the shift.

Note that the station is not being assigned to the crew in phase 1.

**Phase 2:**

After phase 1, the array X[p][h][d] will be fully instantiated and passed to the Phase 2 of the application. Based on the solution from Phase 1, another set of constraint will be handled in Phase2. The decision variables for phase 2 are expressed as booleans:

X[p][h][s] where p is a given person, h is the hour and s is the station that the person has been assigned to for hour h.

Hence, phase 2 decide the station that the crew is assigned to.

**Conversion from Phase 1 to Phase 2:**

Time interval in Phase 1 is defined as half hourly: the demand and supply of crew will be considered for each half-hour interval. The solution from Phase 1 (i.e. X [p][h][d]) and demand will be converted into hourly requirements before passing to Phase 2 so that the performance in Phase 2 can be maintained at an acceptable level.

**Considerations in Phase 1:**

Ensure that the difference between demand (i.e. the number of crew required) & supply (i.e. the number of crew available) of each time interval within allowable gaps (above/below demand). This is the comparison between total number of crew required and total number of crew available for each 1/2 hour (i.e. the summation for all stations of that 1/2 hour).

The following hourly chart shows the number of crew needed varies greatly during the operating hours of the store.

Ensure that the difference between demand & supply of the time interval for each station within allowable gap. Similar to point 1 above, but the constraint here is not cumulative; it applies to each station.

A shift can start and end any time based on the crew's availability. This time flexibility for the shift increases the complexity of the model. (Scheduling is part of the decision process).

Every crew should work only once at any given time interval.

Select those crews first with the preference to work during holiday or public holiday periods if that day being processed is a holiday. This is handled in the objective function using weights for each crew.

demand by giving higher weight to dummy crew. Using the similar technique as in point 3.

Need to keep the total hours and days that the crew has worked for the week, as there is a maximum number of days and hours that a crew can be worked for a week. For example:

Maximum hours for a week: 48
Maximum continuous days for a week: 6

Some parameters that affect the scheduling engine, such as:

Maximum Hours per shift: 8
Minimum Hours per shift: 2

**Considerations in Phase 2:**

The restriction on number of stations that a crew can work during her/his shift needs to be fulfilled. For example, no more than 2 stations for a five hours shift.

The application should assign only one station for each crew for each hour.

Ensure that the difference between demand & supply per hour for each station is within allowable gap (same as phase 1 constraint 4).

All the hours in a shift (defined in phase 1) need to be assigned to a station so that the crew will be working continuously for the entire shift.

The transition constraints. Example: Having a person doing Front Counter, Front Counter, Beverages, Front Counter and Beverages again would be requiring to many changes of stations and should be avoided. A better solution would be Front Counter, Front Counter, Front Counter, Beverages and Beverages. This type of constraint is difficult to handle efficiently in the linear model (too many constraints and booleans need to be introduced). Using an iterative repair method together with the incremental nature of ILOG Planner enable us to handle efficiently the station transition constraints within a shift. An intermediate solution will be produced first without the "station transition" constraints, and we manipulate this imperfect solution to produce an acceptable solution by adding additional constraints to Solver.

A crew will only be assigned to a station if her/his station rating meets the minimum requirement for that station.

Suggest break time for shift longer than 5 hours.

On top of the above constraints in Phase 2, the system also has routine to handle the rotation of station in a shift. For example, if a shift is longer than 5 hours, the system should assign more than 1 station to the crew for that shift.

**Results:**

4000-10000
400-2000
1/2 - 11/2Phase 2 (1 day)3000-100002000-60001 - 2
There is still room for improvement on the response time and more testing will be done to fine tune the model.

**Summary:**

By using Ilog products, the application has produced an optimised solution within a reasonable response time based on the constraints that are defined by the users. The problem was considered difficult for the following reasons:

There are 4 dimensions to this problem: Who should be assigned, When should she/he starts, How long should she/he works, at which position should she/he be assigned.
The level of multi-skilling is very high.
Many difficult constraints to handle (Gradient Type constraints: transition frequency, etc).
Break Policy
The size of the problem.

The Branch and Bound provided by Solver was not really used since particular care was taken to represent the problem as a unimodular matrix.

United Airlines developed a Personnel Scheduling system to schedule its personnel at the airline's reservations offices and airports so as to minimise the cost of providing the necessary service to customers in 1982. Although the details of the model has not been published, but it should be similar to the McDonald's Restaurants Crew Scheduling system as both system also take into account such things as break assignment for each employee scheduled, differences in shift lengths, among other scheduling details. Although with this kind of complexity and huge decision variables, we manage to develop the optimisation engine within three man months using Solver and Planner which we believe it is definitely shorter than the development time of the United Airlines system.

There are other important modules in this application. Such as the training module which keep track of the training details of all crew and also help the store manager to plan the training for the crew. All these we have not talked about them in this paper as they are outside the scope of this paper.

**Appendix 1:**

Company Background:

Accede Computer Solutions Pte Ltd is a Singapore based company with focus on providing turnkey application software using latest Information technology like Client/Server and Internet.

Our customer base comprises both local and overseas corporation such as Jardine Matherson Holding, Havi Food Services (Singapore) Pte Ltd, Brent Asia, The Lion Corporation, Comfort Pte Ltd, D'laser Pte Ltd, Lam Soon Holding, Motorola, National Computer Board's BookNet and etc.

As a company, our mission is to provide world-class application software and to provide superior service to our customers. It is in this light that we have achieved ISO 9001 Certification in November 1996.

Our vision is to be a leading software Consultancy and Development company in Singapore who not only meet international standards in our product quality but also one that attracts good and talented people who will grow with us.

McDonald's Restaurants Pte Ltd, the world renown fast food chain looked for a computer based solution for its Crew Scheduling function a year ago. After considered all the proposal that have been submitted to them, they have selected us based on our expertise on Client/Server and also the excellence service that we have provided to their Distribution Center i.e. Havi Food Services (Singapore) Pte Ltd.

And on our side, in order to generate a good solution on the schedule based on McDonald's requirements, we looked around the market for tools that can help us to achieve this objective. And finally we chose Ilog Solver and Planner as the tool for this project.

This project also serves a long-term objective for Accede Computer Solutions Pte Ltd, as our ultimate aim is to include optimisation modules on distribution in our standard Client/Server application package by using Ilog Planner, Solver and Scheduler.