INTERFACE

by Jack Stout

Computer-Aided What?

It was, I'm sure you will agree, an awkward situation. There I stood, the first day on the job of a new consulting project, when my client showed me a report prepared under an earlier consulting contract by one of my worthy competitors—a huge, international organization of considerable reputation. Unprepared, I was suddenly confronted with something labeled an "EMS System Algorithm," and asked for my opinion.

While I am not without experience in such matters (having worked extensively with telephone protocols, cardiac algorithms, priority dispatch protocols, even elaborate system status management algorithms and similar useful stuff), I had to admit I had never discovered a thing so wonderfully generic or so universally applicable as an "EMS System Algorithm." Here, without modification, is the thing I was asked to comment upon:

$$RT = DD + C \quad \surd A/V^2N \quad (1 + U)$$

Where: RT = Response Time
DD = Dispatch Delay
C = Constant which indicates difficulty of traversing county roads
A = Total County Area
V = Average Vehicle Velocity
N = Number of Response Units
U = Utilization

Using their formula, the consultants had determined a need for 30 ALS units. The client wanted to know, what did I think?

What I *thought* was, "We all have to earn a living, and selling 'EMS System Algorithms' probably isn't any worse than selling bee pollen steeped in bat blood as a cure for brain cancer." What I *said* was, "Hmmm, let me see, that would be something over 5,000 unit hours to handle about a thousand calls a week, 200 of which, according to their estimates, would be transported by ALS units. Should be plenty." Then I bit my tongue to keep from saying any more.

Remember the old joke, "How many (fill in blank) does it take to screw in a light bulb?" This is the same kind of question. How many ALS units it takes to deliver reliable response time performance to a given population depends (heavily) upon a hell of a lot more factors than DD (dispatch delay), C (difficulty of movement), A (area), V (speed), and U (demand). Among other things, the answer depends upon what's written in that blank.

The aim of this article is to describe how computers and formulas can be used—not only to find the answer, but to actually create the answer—and to show when and why your own expert judgment is still your best bet.

1. Demand Pattern Analysis

You don't need a doctorate in statistics or a computer to figure out that more units are needed when call volumes are high and traffic is heavy than when call volumes are low and traffic is light. A computer is, however, required to routinely (e.g., monthly) analyze large volumes of dispatch data and to display the resulting information in a truly useful way.

Computer reports showing the historical frequency of calls for every hour of every day, broken down by run code type and showing the ranges of demand fluctuation each hour, have now become basic planning tools in high-efficiency ALS systems.

Computer-generated maps, often nothing more than coded markings with justification marks to allow use of a transparent overlay showing major streets, graphically display the geographic distribution of these very same calls. These 168 maps (one for each hour of every day in a week), when combined with the reports showing both typical call volumes and historical demand fluctuations, furnish a remarkably clear and often surprising picture of where and when the action is, and how often these patterns tend to repeat themselves. This is fundamental information, with the only safe alternative being a shotgun approach to deployment—i.e., lots of units, all the time, everywhere.

If you've ever tried to produce this type of information on a regular basis without a computer, then you don't need me to tell you that this is clearly a job for computers.

2. Developing Your Initial System Status Plan

System status management (SSM) is our industry's term for the strategies and tactics used to continuously manage the resources available in the system at any point in time so as to anticipate and prepare for the very next call. (See "System Status Management," May 1983 *jems* for a more detailed discussion of SSM practices.) In our industry's most reliable and efficient systems, SSM skills are highly refined.

In its most basic application, however, SSM deals with the around-the-clock and geographic distribution of on-duty crews and their planned redeployment depending upon time of day, day of week, and number of units available for dispatch at any point in time. For example, there will be a system status plan (SSP) specifying where the system's last three available units should be positioned if the time is between 4 and 5 p.m., the day is Thursday, and the fourth unit in the system has just been dispatched to another call.

To put this in perspective, consider a medium-sized ALS system which operates with as few as eight ALS units

---

Jack Stout has been at the forefront of innovations in the design and implementation of EMS systems for the past dozen years. If you have a question, a problem, or a solution related to the public/private interface in prehospital care, address your letter to "Interface" jems, *P.O. Box 1026, Solana Beach, CA 92075.*

during off-peak periods and as many as 18 units at peak load demand. Assuming the overall coverage level averages 13 ALS units, activities will be governed by approximately 2,184 separate deployment plans (i.e., 13 units × 24 hours × seven days in the week). Sounds like these folks are going to an awful lot of work just to be efficient, doesn't it? They are, and that's just the beginning.

Theories aside, the most effective *initial* SSPs developed to date were developed by people—not by computers. I believe that may always be program could be developed, computerizing the prerequisite data would be extremely expensive, and in any given system the program would be used only once.

No matter how refined such an algorithm would get, the resulting SSP would soon be modified using the far superior refinement techniques discussed below. Experience has shown that, using the computer-generated reports discussed previously, combined with a short-term increase in coverage levels, your most experienced dispatchers and street people can produce a safe and functional initial SSP that can then be refined using far superior methods.

Thus, while computers are useful, even essential, for analyzing the data and displaying the information needed to develop an effective initial system status plan, nothing can top informed human judgement for developing the plan itself.

System Status Management

Here's where the computer can really come into its own. Since I wrote the 1983 *jems* article on SSM practices, great progress has been made in automating system status plans and SSM controls. While several good SSM-based computer aided dispatching (CAD) systems do exist as of this writing, perhaps the most advanced SSM/CAD in our industry is the one now being used in Fort Worth, Texas.

Now for a confession. I usually stay as far away as I can get from the data processing departments of local government. Done right, development of an SSM/CAD is far more complex than that of an ordinary CAD system, and well beyond the practical capabilities of most local government DP departments. So when J. Marshall, head of Fort Worth's data processing department, asked for a shot at expanding their fire-based CAD to handle SSM controls, I politely but firmly advised against it several times.

Finally, Marshall took me aside and said he fully understood my position but that this was an unusual situation, and it sure was. My entire contribution to the effort was participation in two early meetings and a two-page partial description of what an SSM/CAD should be able to do. Taking it from there, and in cooperation with Fort Worth's ALS contractor, Medstar, Marshall's shop turned out a fully functional SSM/CAD second to none, in record time.

How good is it? It's so good that upon its completion, Medstar agreed to allow its contractual response time commitment to be changed from 10 minutes maximum on not less than 90 percent of all calls to eight minutes maximum on the same percentage, *without* increase in rates or subsidy. It's that good a tool. (In a city of that size, the cost of such an upgrade in service achieved any other way would easily exceed half a million dollars a year.)

INTERFACE

There isn't space here to describe the extensive capabilities of an SSM/CAD. At its simplest level, it "remembers" the SSPs for every hour of every day at every level of remaining production capacity, monitors the changes in status and movements of every unit working in the system, displays the current status and location of every available unit, recommends redeployment whenever the system is out of compliance with its planned configuration, records and time-stamps the activities of every unit in the system (assigned to runs or not), and organizes that data into primary data files for later analysis in a batch-oriented environment. Those are the basics. Fort Worth's system does all that and much more and is scheduled for still more enhancements in coming months.

My own opinion is that it becomes impossible to reliably handle SSM controls on a manual basis when your peak load coverage level exceeds seven or eight units. After that level, you need automation.

3. Fine-Tuning the SSP

You can't fine-tune an SSP that you haven't been using. Put another way, if you develop an SSP but deviate from it regularly, then when problems occur you can never be certain whether they were the result of following the SSP (meaning the plan needs adjustment), or of failing to follow the SSP (meaning the plan might work but you'll never know).

Whereas the process of developing the initial SSP was largely deductive, the process of fine-tuning the SSP is almost entirely experimental. Here again, the computer can help us analyze data and display information in a useful form. The computer cannot, however, select the objectives of our fine-tuning efforts.

A. Objectives of Fine-Tuning the SSP. Tracking down and destroying response time problems is the best and most important objective of SSP refinement. It is not, however, the only objective. Here are some others that can come into play:

- Reduce non-emergency service delays.
- Equalize service among neighborhoods.
- Safely "make room" for more non-emergency service production at low marginal cost.
- Reduce the use of on-call crews.
- Reduce the frequency of post-to-post moves.
- Equalize workloads among crews.
- Differentiate workloads of 24-hour crews from those of short-shift crews.
- Furnish better mutual aid service.
- Reduce own use of mutual aid service.
- Cut overtime.
- Employ schedules more convenient to crews.
- Battle cream-skimmers working your market.
- Cut production costs without hurting response time performance.

B. Isolating Response Time Problems. Computers can help identify and isolate response time problems. There is no such thing as a "general" response time problem; they occur during some time periods and not during others. During a given period of time, they occur more in certain geographic areas than in others. Response time problems must be solved when and where they are happening, so unless you can afford the shotgun approach, you've got to know exactly when and where your response time problems are taking place.

Remember the 168 maps discussed in Section I? By changing the coding on late runs being displayed, you can pinpoint the hours when problems are happening and the specific geographic area or areas involved. Then you can get rid of the shotgun and pick up your rifle.

C. Diagnosing Response Time Problems. Knowing when and where your late runs are happening, you'll now need to find out why they are happening. Of the thousands of late-run incident reports I have analyzed (from a number of different systems), the majority of causes could not be corrected by adding more units. To fix what's wrong, you must know what's wrong. To know what's wrong, your control center personnel will have to faithfully complete a detailed late-run incident report on every run resulting in an extended response time. In my opinion, the computer can't help you here. A few of the most common causes of response time failure are:

- Unit got lost.
- Bad (or no) routing instructions.
- Road construction that could have been known in advance and avoided with routing instructions.
- Dispatch center overload.
- Failure to follow the SSP.
- Unit-alert communications failure.
- Equipment failure en route.
- Stupid shift change timing.
- Slow shift change procedures.
- Crew not present at reported location.
- Predictable traffic congestion not accounted for in the SSP.
- Slow pickup/dropoff procedures at one or more facilities.
- Bad out-of-chute time.

Once you know exactly when, where, and why your response time problems are happening, you are ready to begin fine-tuning your SSP—but only if you've been following it. Computers can help you pinpoint your problem hours and problem neighborhoods. By showing you hours and areas where you never experience late runs, the computer can also help you locate excess production capacity that can be shifted to other times and/or places.

Unfortunately, there is still no substitute for sitting down and studying the late-run incident reports for every poor response that took place during a particular problem period. Because many of the causes of poor response time performance cannot be solved by simply adding more units, changing post locations, or changing post priorities, and because judgment must come into play when selecting the objectives of the fine-tuning effort (many of which will compete with each other), the computer cannot tell you how best to adjust and refine your SSP.

As our industry matures, computers are playing an increasingly important role in improving, often simultaneously, the economic efficiency response time reliability of ALS systems. Now and in the future, the secret of successful automation in EMS lies in knowing when computers can help, and when your human judgment must prevail. ☐