Pergamon

0969–6016(93)E0005–D

Int. Trans. Opl Res. Vol. 1, No. 3, pp. 353–361, 1994
Elsevier Science Ltd.
0969-6016/94 $7.00 + 0.00

Automated Manpower Rostering: Techniques and Experience

C. M. KHOONG, H. C. LAU and L. W. CHEW
National Computer Board, Singapore

We present *ROMAN*, a comprehensive, generic manpower rostering toolkit that successfully handles a wide spectrum of work policies found in service organizations. We review the use of various techniques and methodologies in the toolkit that contribute to its robustness and efficiency, and relate experience gained in addressing manpower rostering problems in industry.

Key words: allocation, artificial intelligence, heuristics, optimization, scheduling, search, travelling salesman

INTRODUCTION

Manpower is a highly critical resource in many industries. It is not surprising that organizations place substantial investment into manpower management matters. Manpower rostering, which addresses the issues of employing, allocating and scheduling manpower resources to meet operational demands, is at the heart of the manpower management cycle. Rostering is a particularly prominent activity in service organizations where employees need to be scheduled across several workshifts to handle varying customer demands 24 hr a day and seven days a week.

The generation of rosters is a complicated process of meeting demands with available resources subject to a variety of work policy constraints and goals. The need for automated manpower rostering systems is thus intuitive and appealing. The system would reduce time spent on creating and modifying rosters for employees, as well as improve on the quality of the rosters, by optimizing manpower employment and deployment costs. Through our contacts, we are aware of some organizations in which a pool of manpower planners spend several days producing each new set of rosters manually, and yet dissatisfactions with the quality of work schedules are routinely expressed by the management and employees.

Manpower rostering has been actively researched in the operations research and management science communities for a long time. Popular domains studied in the literature include hospital nurses (Kostreva and Jennings, 1991; Miller *et al.*, 1976; Smith and Wiggins, 1977; Warner, 1976), bus drivers (Martello and Toth, 1986; Rousseau, 1984), telephone operators (Henderson and Berry, 1976; Keith, 1979), and airport ground crew (Chew, 1991; Holloran and Bryn, 1986). The broad spectrum of domains has led to several distinct solution approaches in the literature, each tailored to a particular class of problems. However, comparatively little work has been done on general models.

In this paper, we present *ROMAN*, a generic toolkit for manpower rostering, developed at the Information Technology Institute in Singapore. The toolkit is a culmination of a year-long applied research effort. Unlike extant systems, *ROMAN* is designed to address a very wide spectrum of work policies encountered in service organizations. Techniques originating from artificial intelligence, operations research and software engineering arenas are combined into a toolkit that is highly flexible, robust, efficient, customizable and extensible. To date, the toolkit has been licensed to major service organizations in the local health care and transportation sectors.

This paper discusses *ROMAN* in terms of its architectural structure and issues related to its industrial implementation. Analytical modelling and algorithmic aspects in *ROMAN* will be reviewed only briefly here, since these aspects have been explained in depth in previous publications of the authors (Chew *et al.*, 1991; Khoong, 1991; Khoong, 1992; Khoong and Lau, 1992). The rest of this paper is organized as follows. In the next section, a generalized view of manpower rostering models is developed. This provides the basis for the discussion of the *ROMAN* framework in the

Correspondence: C. M. Khoong, Information Technology Institute, National Computer Board, 71 Science Park Drive, Singapore 0511

following section, where the innovative contributions of the toolkit are highlighted. Following that, the scheduling techniques employed in the toolkit are reviewed. We then highlight issues on successful industrial implementation drawn from our experience in the service industry. Finally, we conclude with some remarks.

MANPOWER ROSTERING MODELS

In the overall manpower planning flow in a service organization, rostering is a highly visible intermediate-term activity. On one hand, it implements long-term organizational objectives, and on the other hand produces direct impact on the day-to-day running of operations (i.e. deployment). Figure 1 depicts the flow of manpower planning from long-term to intermediate-term, and thereon to day-to-day functions.

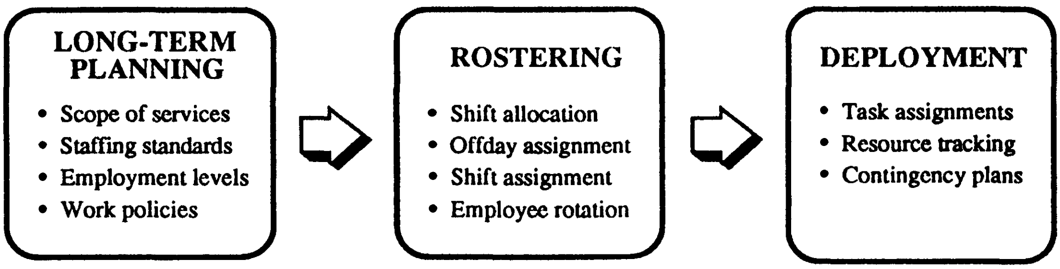


Fig. 1. Manpower planning flow.

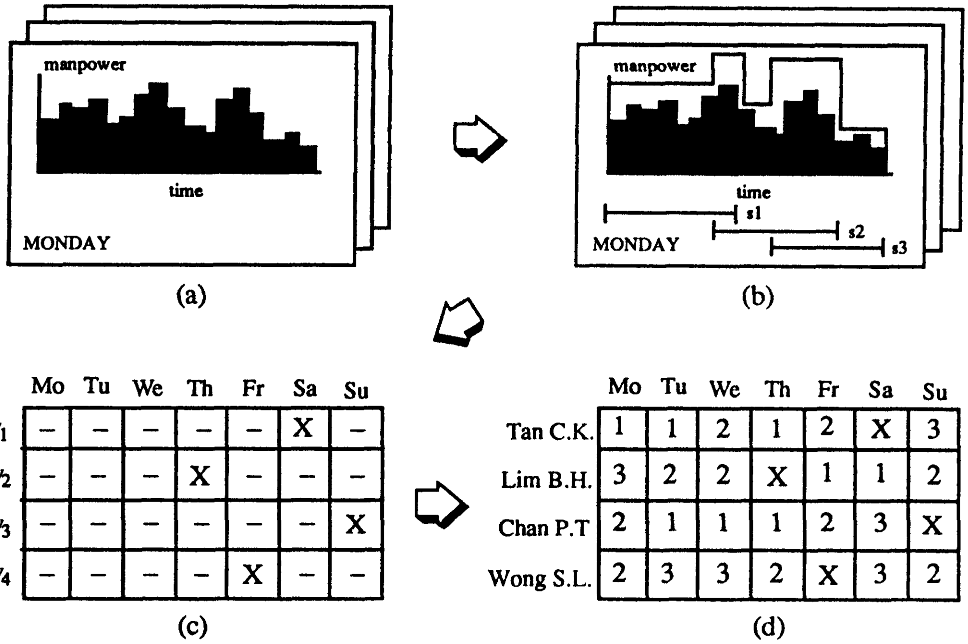
We mention here some basic terms of reference that will be used throughout this paper. A *planning period* is the temporal period for which manpower planning is done, e.g. one week, starting Monday. An *employee* is a basic unit of resource in the planning and scheduling process; in real terms an employee may actually be a team rather than an individual. A *workshift* (or simply *shift*) is a period of time when an employee will perform work within a workday. An *offday* is a day when an employee will not perform any work. A *shift pattern* is a contiguous chain of workshift and offday assignments for an employee in a planning period stretch. A *roster* is a set of shift patterns that apply to the same planning period stretch.

Known approaches to solving manpower rostering problems may be distinguished into several distinct models. Each of these models falls into either one of two design modes, namely *cyclic* or *individualized*. We explain each of these modes below.

A cyclic roster consists of a set of master shift patterns that is rotated across the employees over time. For example, consider a planning period of one week. Let W be the number of employees rostered. Define shift pattern p as the chain of workshifts and offdays for the current week of employee p, where p ranges from 1 to W. In the next week, employee p is assigned shift pattern $p + 1$ for $p < W$, and employee W is assigned shift pattern 1.

An individualized roster consists of shift patterns that are unique for each employee. A new set of shift patterns is generated for each planning period stretch. The shift pattern assigned to an employee for the next planning period stretch depends only on his shift patterns for the current and previous planning period stretches, preferences pertaining to him, and the demand constraints. Preferences may be specified as either permanent or ad hoc bids for or against particular types of duties. Permanent bids are specified for particular days of the week, while ad hoc bids are specified for particular calendar dates.

Note that it is not meaningful to apply both cyclic and individualized design modes to the same roster. Each design mode has its own advantages and range of applications. Cyclic rosters are easy to manage and are more stable, but also more resistant to changes in demand levels. Individualized rosters, on the other hand, can cater to the preferences of individual employees and adapt to fluctuations in demand levels more readily, but are more tedious to generate. Examples of work on cyclic scheduling are found in (Baker and Magazine, 1977; Burns and Koop, 1987; Emmons, 1985; Emmons and Burns, 1991; Koop, 1988; Lowerre, 1977; Panton, 1991). Individualized scheduling is



(c)

	Mo	Tu	We	Th	Fr	Sa	Su
w_1	–	–	–	–	–	X	–
w_2	–	–	–	X	–	–	–
w_3	–	–	–	–	–	–	X
w_4	–	–	–	–	X	–	–

(d)

	Mo	Tu	We	Th	Fr	Sa	Su
Tan C.K.	1	1	2	1	2	X	3
Lim B.H.	3	2	2	X	1	1	2
Chan P.T	2	1	1	1	2	3	X
Wong S.L.	2	3	3	2	X	3	2

Fig. 2. Core stages in the general rostering model. In this illustration, the planning period is one week, starting Monday, and there are three workshift types. Given the temporal demand histograms (a), the shift allocation stage computes the staffing requirements (b) for each shift in each day. The supply histogram is superimposed over the demand histogram. An employment level (4 in this case) is derived. Let the employees be indicated by w_1, w_2, w_3 and w_4. The offday scheduling stage (c) marks out the offdays (labeled 'X') on the roster template. The workshift scheduling stage (d) assigns workshifts (labelled '1', '2', or '3'). For cyclic scheduling, real employee names are assigned when the roster is despatched for each planning period strength. For individualized scheduling, the names may be pegged to the rows of the roster permanently for the purpose of ensuring fairness over planning period stretches.

advocated in (Burns and Carter, 1985; Kostreva and Jennings, 1991; Miller *et al.*, 1976; Smith and Wiggins, 1977; Warner, 1976).

It is a popular practice to decompose the general rostering problem into three core stages, namely *shift allocation*, *offday scheduling* and *workshift scheduling*. This decomposition applies to both cyclic and individualized design modes. These stages are illustrated in Fig. 2. Basic mathematical programming formulations for the core stages are studied in (Baker, 1976) and (Tien and Kamiyama, 1982).

Shift allocation is concerned with the determination of staffing levels for each shift in each day. Allocation is dependent on the set of feasible shift types and the temporal demand profile. For planning periods defined to be more than a day in length (e.g. one week), it is a common practice to solve the shift allocation problem separately for each day, since shifts generally fall within a single workday. Shift allocation is sometimes coupled with the problem of employment level determination, i.e. computing the minimum number of employees needed to fulfill staffing demands over the planning period stretch.

Offday scheduling is concerned with the generation of offday assignments in the roster, subject to such constraints as daily staffing demands and workstretch and offstretch constraints. Other criteria may be the maximization of off-weekends and fairness in assignments across the offday patterns.

Workshift scheduling is concerned with the generation of specific workshift assignments in the roster, subjected to the staffing level for each shift in each day, shift change constraints and offday assignments. Other criteria may be constraints on total working time per planning period and fairness in assignments across workshift patterns.

THE *ROMAN* TOOLKIT

All known work on manpower rostering assumes that only a limited set of contraints needs to be handled (e.g. staffing and workstretch constraints); in fact, most of them also assume specific

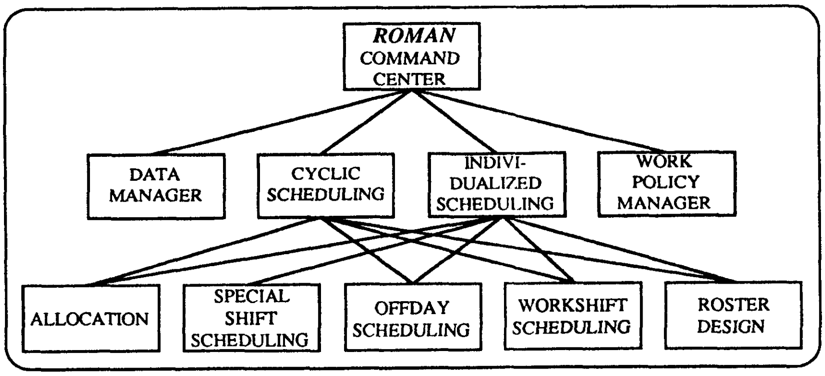


Fig. 3. The *ROMAN* framework.

parameters for the constraints considered (e.g. workstretches between five and seven days). It also appears that none of the extant systems offer a true integration of all the major models along both the design mode and core stage dimensions.

The *ROMAN* framework, on the other hand, is an integration of the major models discussed in Section 2. Both cyclic and individualized design modes are catered for, and in each design mode all three core stages are handled. The *ROMAN* framework also provides several other supporting functionalities that makes the toolkit comprehensive, robust and flexible. A graphical representation of the framework is shown in Fig. 3.

Technical details on *ROMAN* have been reported elsewhere (Chew *et al.*, 1991; Khoong, 1991; Khoong, 1992; Khoong and Lau, 1992); our purpose in this paper is to present a complete view of the toolkit architecture and to emphasize the innovative contributions of *ROMAN* to industrial manpower rostering applications.

Figure 3 shows the integration of both cyclic and individualized design modes and the incorporation of all the core stages for each mode in the toolkit. But the conceptual differences and operational similarities of each design mode are managed and exploited to streamline *ROMAN*'s functionalities and to present a uniform interface to users of each mode.

The only operational differences between the two design modes are that cyclic scheduling does not include Special Shift Scheduling or the handling of bids. Both design modes require a master specification file, an employee data file and sometimes supplementary specification files. In individualized scheduling, the employee bids file is also expected. In terms of system outputs, the look and feel of the Offday and Workshift Scheduling submodules differ because names are already pegged to the roster in individualized scheduling, but not yet in cyclic scheduling (until the Roster Design stage).

The Data Manager handles database interface requirements. Default flat file formats are assumed, which users may replace with more sophisticated database interfaces if necessary. The Work Policy Manager is a mechanism to access and manipulate the work policy. A translator for an English-like specification language is provided for this purpose. The user feeds the work policy to *ROMAN*, using this language, in a specification file. The toolkit also provides a simple command language as a default user interface. A more sophisticated user interface layer may be added if necessary.

The Allocation submodule computes shift staffing levels from temporal demand histograms. The Special Shift Scheduling submodule generates assignments for a particular workshift type that are differentiated from the normal workshift types. These arise in individualized scheduling due to consideration for fairness across employees. The Offday Scheduling submodule generates offday assignments, taking into account staffing needs. The Workshift Scheduling submodule generates normal workshift assignments to fulfill the shift staffing requirements. The Roster Design submodule converts the working roster (i.e. the roster that is manipulated by the other submodules) into an operational one, and facilitates fine-tuning of the roster. (A roster is said to be operational if all the information that is necessary for the roster to be despatched for real use is attached.)

The key advantages of the scheduling functionalities in *ROMAN* are their efficiency and support for interactive and flexible usage. The specification language facility is innovative in its generality and simplicity. The default database interface and command language are simple but useful mechanisms for rapid integration of the toolkit into the information systems of client organizations.

The features of the *ROMAN* toolkit may be summarized as follows:

- Implementation is entirely in the standard C programming language.
- Comprehensive coverage of work policy constraints, including: shift length, rest length, shift change, offday preference, offday stretch, work stretch, working time.
- Comprehensive mechanisms for users to define the calendar structure, workday structure, workshift types, offday types and manpower cost measures.
- Flexible facilities for decomposing the organizational workforce structure into departments and skill groups.
- Cyclic as well as individualized design modes are handled.
- Optimizes on all aspects of quality, including deployment cost, staffing level adequacy and employee welfare, while respecting tradeoffs desired by the user.
- A simple, flexible and yet powerful specification language for users to specify organizational policies and constraints.
- An easy-to-use command language that allows the user to interact directly with the toolkit as well as to integrate the toolkit into delivery systems.
- Beyond routine rostering, also facilitates effective what-if analyses on new work policies, employment levels, demand levels, deployment profiles.
- No hardware, operating system, or software tool dependencies.
- Low computation time and memory space requirements.
- High maintainability, customizability and extensibility.

The authors are not aware of any other existing manpower rostering system that has a comparable coverage of constraints, type definitions, or supporting facilities. The toolkit has been successfully tested on a variety of platforms, including DEC VAX minicomputers, SUN Sparc workstations and IBM PCs. For rosters with about 50 employees and about 10 shift types, the entire roster generation process may be completed in a few sec on a workstation. The problem sizes that the toolkit can handle are limited only by the amount of run-time memory available on the delivery platform. Controls are implemented that allow the user to trade off solution optimality with computational efficiency when the problem size is large.

SCHEDULING TECHNIQUES IN *ROMAN*

In this section, we review very briefly the techniques currently used to implement *ROMAN*'s scheduling functions. The general algorithmic framework is drafted in (Khoong, 1991). The first phase of development work on *ROMAN* was focussed on cyclic scheduling, and this work is reported in (Khoong and Lau, 1992). The workshift scheduling algorithm in *ROMAN* has since gone through some extensions, which are reported in (Khoong, 1992). The second phase of development work on *ROMAN* extended the toolkit to handle individualized scheduling, in a way that is uniform with respect to the general framework. This phase is reported in (Chew *et al.*, 1991).

In the rest of this section, we focus on the Allocation, Offday Scheduling and Workshift Scheduling submodules. The bulk of the algorithmics in *ROMAN* are found in these submodules. Less substantial computation is found in the Work Policy Manager and the Special Shift Scheduling and Roster Design submodules. The Work Policy Manager has some intelligence to generate the shift change cost matrices, and even shift types and offday pattern types, if required, from constraints specified in the work policy. The Special Shift Scheduling submodule may be seen as a simplification of the individualized workshift scheduler (to be described below), but augmented with a so-called 'probability ruler' concept to randomly distribute shift assignments in a fair manner. The Roster Design submodule supports interactive modifications of shift assignments in the roster. The submodule performs consistency checks against constraints and highlights problems (such as employment level shortages and shift change constraint violations) arising from the modifications.

The Allocation submodule models the computation of daily shift staffing levels from temporal demands as a *set covering* problem, in which the objective is to determine the minimum-cost set of shifts which would adequately cover demands for manpower. Given the intractability of set covering, we implemented a greedy set covering heuristic with a good performance bound. Another heuristic is run to further trim the staffing levels output by the set cover routine if possible, by checking for shift types that can do with lower turnouts or that can be substituted by cheaper shift types. From the final shift staffing levels, an employment level is suggested, which is computed based on the minimum number of staff required given the staffing profile over the planning period, offday requirements and constraints arising from the total number of hours that an employee may work in a planning period.

The Offday Scheduling submodule has two implementations, one for the cyclic mode and another for the individualized mode. Cyclic offday scheduling is modelled as a process of selecting offday patterns (i.e. planning period stretches with some days designated off) followed by a process of arranging the offday patterns to form a cycle. A candidate cycle is generated for each offday pattern type as a starting point in the cycle, and the best cycle is returned as the offday schedule. The generation of each cycle involves a branch and bound algorithm that respects staffing requirements, offday preferences, workstretch and offstretch constraints, and spreading of offday patterns of the same type across the roster. The output of the branch and bound algorithm is further passed to a 2-*opt* iterative improvement procedure, in which the cycle is treated like a *travelling salesman tour* that may be improved via local search.

The individualized offday scheduler runs in two stages. The first stage assigns weekends off using a greedy algorithm. The second stage, for assigning weekdays off, is a *transportation* problem complicated by side constraints. We implemented a backtracking algorithm that minimizes employee shortage and balances the assignment of individual offdays over the days of the planning period in light of staffing levels. Both stages respect considerations as in the cyclic offday scheduler as well as fairness across employees (given the current roster and the history of past offday assignments) and employees' bids for offdays.

The Workshift Scheduling submodule also has separate implementations for the cyclic and individualized modes. The cyclic workshift scheduler first uses a fast heuristic to generate an initial, complete roster. The heuristic uses the idea of *monotonic* shift change sequences to derive 'constrainedness' of shift types, which in turn provides the basis for planting workshifts into the roster day-wise. The output of the heuristic is further subject to 2-opt iterative improvement on the assignments for each day of the planning period. The resulting roster, which may already be quite good, is then used as an initial upper bound for a more complicated branch and bound algorithm that aims at global optimality. 'Toggles' are provided for the user to exert his influence in the rostering process; these toggles also translate into pruning mechanisms in the branch and bound search.

The individualized workshift scheduler models the assignment process as a series of *weighted bipartite graph matching* problems, one for each day of the planning period. The Hungarian method is used as the algorithm. Each day-wise matching process considers the characteristics of the assignments to each individual so far in the roster to match the individuals to the shifts required for the next day. The various considerations, such as incurred shift cost and incurred shift change cost, may be weighted by the user to influence the quality of the overall workshift assignment.

The Allocation, Offday Scheduling and Workshift Scheduling submodules are all designed to run in interactive mode, in support of what-if analyses and interactive optimization. The user can change policies, constraints and system outputs at each stage of the rostering process and examine the corresponding effects.

ISSUES IN INDUSTRIAL IMPLEMENTATION

The success of a manpower rostering system depends on a host of factors. We distinguish the following factors: power, efficiency, ease of use, flexibility, maintainability, extensibility, portability and impact. All of these factors are prominent concerns in the design of *ROMAN*, and we also strongly recommend consideration for them in the development of manpower rostering systems in general. We discuss each factor below.

By *power* we mean the comprehensiveness of the system, i.e. the range of work policy constraints and objectives that may be handled by the system. Most systems are based on models that make certain assumptions. These assumptions may exist in order to reduce the complexity of the problem, or because they hold for the target users. It is also often the case that subjective criteria exist that cannot be incorporated into the automated rostering process, especially for matters related to employee welfare. System developers should make clear and honest statements on the scope of system functionalities, and provide facilities that will (probably through man–machine interactive computation) help to make up for the deficiencies of the system.

By *efficiency* we mean the speed with which the system can derive solutions. Early manpower rostering systems that are based on integer programming models tend to be slow, taking hours of run-time to derive a single solution. The decomposition of the general rostering problem into several core stages helps in reducing problem complexity, but each stage remains computationally difficult. In a what-if analysis situation, the user cannot afford to wait too long each time some parameter is perturbed. Therefore, the need for fast heuristics with good solution quality is clear. Furthermore, users may prefer to have the system run on low-cost personal computers. This places even more emphasis on efficiency considerations.

Ease of use is an obvious concern, yet often poorly handled, because the system developer may not have invested sufficient effort in understanding the needs, expectations and psychology of the users. The internal data structures in the system may hold large amounts of information on employees, staffing demands and a collection of work policy constraints, many of which would typically be in numerical matrix forms. Unfortunately, the information is meaningless to the user in these internal formats. For instance, it would not be reasonable to expect the user to specify shift change constraints to the system in the form of a numerical matrix. Some form of interface is needed for the user to specify constraints and other pertinent aspects of the work policy (unless the work policy is already hardwired into the computer codes, which may be a bad design decision).

Flexibility has many facets. In the context of manpower rostering, a key flexibility criterion is the user's ability to exert his influence in the rostering process in ways that, together with the system's computations, would achieve rosters that are acceptable to the users. Man–machine interactive computation is an intuitive but challenging means to achieve good solutions. The areas where dynamic human intervention is possible should be identified, and inputs from the user should be incorporated smoothly. The interactive scheduling facilities of *ROMAN* are a step in this direction.

Maintainability is critical for the viability of systems. A manpower rostering system would fail to stand the test of time if it is unmaintainable. This is especially relevant to efforts directed by researchers with little experience in building industrial systems, who tend to ignore the need for proper system documentation and extensible programming style. The manpower rostering needs of an organization are likely to change over time, given the continually increasing sophistication of the workforce and the evolution of organizational goals and policies. Systems that are hardwired to handle a very restrictive set of parameters are likely to be thrown away by users once the deficiencies are detected.

Extensibility is important because it would be difficult for a generic system to be fully comprehensive across all client organizations. The concerns of manpower planners, apart from the central issue of rostering full-time manpower, also include provision for part-time and temporary staff, demand histogram massaging, employee tracking for the purposes of recall and standby, etc. Failure to give consideration to some of these peripheral issues may actually lead to disuse of the system if the manpower planner feels that a substantial portion of his responsibilities cannot be alleviated through automation.

Portability is always cited as being desirable, but seldom practised in the development of manpower rostering systems. This is probably due to the fact that most of the development efforts were targeted at one specific organization, without regard for the possibilities of marketing the systems to other organizations. Portability is crucial for toolkits such as *ROMAN* to achieve widespread use among service organizations.

The issue of *impact* is quite subtle. Automation is not always a good word, and it is important to help users assess their manpower rostering needs carefully. From our experience with users in the local industry, it appears that a large proportion of users do not understand what rostering systems can do for them. Some of these organizations have highly static demands and/or peculiar work policies that make rostering easily doable by hand. One such class of peculiarities is studied in (Khoong, 1993). Other organizations may have extremely complex and subjective constraints in their work policies that render automation impossible. In this situation, the users should be made to understand that no useful impact would be derived from automation. There may also be organizations in which rostering decisions are dominated by deployment considerations, such that the separation of rostering from deployment planning may be unclear. In this case, it may be sensible to advise the user on the planning methodology first before considering automation for rostering. It may be the case that rostering is actually not needed!

CONCLUDING REMARKS

Manpower is an increasingly expensive and strategic resource in today's service industry. Automation of manpower planning functions will therefore continue to be a critical management concern. The *ROMAN* toolkit can therefore be seen as a timely and innovative contribution to automated manpower rostering needs in industry.

Numerous useful extensions to our work are envisaged. From our industrial contacts, it is clear that industry also lacks tools for both long-term manpower planning and daily task deployment functions. Extant systems for long-term manpower planning are mostly restricted to data management capabilities on limited hardware platforms with little or no analytical functionality. Current manpower deployment systems, to our knowledge, are all tailored to the needs of specific user organizations with no consideration for generalization. A generalized, integrated system concept encompassing long-term manpower planning, manpower rostering and manpower deployment is a highly attractive and challenging concept. We are currently pursuing the possibilities of such a concept.

REFERENCES

Bailey, J. (1985). Integrated days off and shift personnel scheduling. *Computers & Industrial Engineering*, Vol. 9, pp. 395–404.

Baker, K. R. (1976). Workforce allocation in cyclical scheduling problems: a survey. *Operational Research Quarterly*, Vol. 27, pp. 155–167.

Baker, K. R., Burns, R. N. & Carter, M. (1979). Staff scheduling with day-off and workstretch constraints. *AIIE Transactions*, Vol. 11, pp. 286–292.

Baker, K. R. & Magazine, M. J. (1977). Workforce scheduling with cyclic demands and day-off constraints. *Management Science*, Vol. 24, pp. 161–167.

Balakrishnan, N. & Wong, R. T. (1990). A network model for the rotating workforce scheduling problem. *Networks*, Vol. 20, pp. 25–42.

Brownell, W. S. & Lowerre, J. M. (1976). Scheduling of work forces required in continuous operations under alternative labor policies. *Management Science*, Vol. 22, pp. 597–605.

Burns, R. N. (1978). Manpower scheduling with variable demands and alternate weekends off. *INFOR*, Vol. 16, pp. 101–111.

Burns, R. N. & Carter, M. W. (1985). Work force size and single shift schedules with variable demands. *Management Science*, Vol. 31, pp. 599–607.

Burns, R. N. & Koop, G. J. (1987). A modular approach to optimal multiple-shift manpower scheduling. *Operations Research*, Vol. 35, pp. 100–110.

Chew, K. L. (1991). Cyclic schedule for apron services. *Journal of the Operational Research Society*, Vol. 42, pp. 1061–1069.

Chew, L. W., Khoong, C. M. & Lau, H. C. (1991). A generic framework for individualized manpower rostering. *Proceedings of Workshop on Optimization* '91, (Singapore, December 13), pp. 13–23.

Emmons, H. (1985). Work-force scheduling with cyclic requirements and constraints on days off, weekends off, and work stretch. *IIE Transactions*, Vol. 17, pp. 8–16.

Emmons, H. & Burns, R. N. (1991). Off-day scheduling with hierarchical worker categories. *Operations Research*, Vol. 39, pp. 484–495.

Glover, F. & McMillan, C. (1986). The general employee scheduling problem: an integration of MS and AI. *Computers & Operations Research*, Vol. 13, pp. 563–573.

Henderson, W. B. & Berry, W. L. (1976). Heuristic methods for telephone operator shift scheduling: an experimental analysis. *Management Science*, Vol. 22, pp. 1372–1380.

Holloran, T. J. & Bryn, J. E. (1986). United airlines station manpower planning system. *Interfaces*, Vol. 16, pp. 39–50.

Keith, E. G. (1979). Operator scheduling. *AIIE Transactions*, Vol. 11, pp. 37–41.

Khoong, C. M. (1991). Decision support systems for manpower planning in service industries. *Proceedings of the International Conference on Productivity* '91 (Singapore, 14–16 August), pp. 305–316.

Khoong, C. M. (1993). A simple but effective heuristic for workshift assignment. *OMEGA*, Vol. 21, pp. 393–395.

Khoong, C. M. (1993). The chain rostering problem. (Submitted.)

Khoong, C. M. & Lau, H. C. (1992). *ROMAN*: an integrated approach to manpower planning and scheduling. In O. Balci, R. Sharda and S. A. Zenios (Eds) *Computer Science and Operations Research: New Developments in Their Interfaces*. Oxford: Pergamon Press, pp. 383–396.

Koop, G. J. (1988). Multiple shift workforce lower bounds. *Management Science*, Vol. 34, pp. 1221–1230.

Kostreva, M. M. & Jennings, K. S. B. (1991). Nurse scheduling on a microcomputer. *Computers & Operations Research*, Vol. 18, pp. 731–739.

Love, Jr, R. R. & Hoey, J. M. (1990). Management science improves fast-food operations. *Interfaces*, Vol. 20, pp. 21–29.

Lowerre, J. M (1977). Work stretch properties for the scheduling of continuous operations under alternative labor policies. *Management Science*, Vol. 23, pp. 963–971.

Martello, S. & Toth, P. (1986). A heuristic approach to the bus driver scheduling problem. *European Journal of Operations Research*, Vol. 24, pp. 106–117.

Miller, H. E., Pierskalla, W. P. & Rath, G. J. (1976). Nurse scheduling using mathematical programming. *Operations Research*, Vol. 24, pp. 857–870.

Morris, J. G. & Showalter, M. J. (1983). Simple approaches to shift, days-off, and tour scheduling problems. *Management Science*, Vol. 29, pp. 942–950.

Panton, D. M. (1991). On the creation of multiple shift continuous operation cyclic rosters under general workforce conditions. *Asia-Pacific Journal of Operational Research*, Vol. 8, pp. 189–201.

Rousseau, J.-M. (1984). Crew scheduling methods in the operation and planning of transportation systems. In M. Florian (Ed.) *Transportation Planning Models*. Amsterdam: North-Holland, pp. 439–472.

Smith, L. D. & Wiggins, A. (1977). A computer-based nurse scheduling system. *Computers & Operations Research*, Vol. 4, pp. 195–212.

Tien, J. M & Kamiyama, A. (1982). On manpower scheduling algorithms. *SIAM Review*, Vol. 24, pp. 275–287.

Warner, D. M. (1976). Scheduling nursing personnel according to nursing preference: a mathematical programming approach. *Operations Research*, Vol. 24, pp. 842–856.