L

Lack of Memory

- ▶ Exponential Arrivals
- ▶ Markov Processes
- ▶ Markov Property
- ▶ Memoryless Property
- ▶ Poisson Process
- ▶ Queueing Theory

Lagrange Multipliers

The multiplicative, linear-combination constants that appear in the Lagrangian of a mathematical programming problem. They are generally dual variables if the dual exists, so-called shadow prices in linear programming, giving the rate of change of the optimal value with constraint changes, under appropriate conditions.

See

- ▶ Lagrangian Function
- ▶ Nonlinear Programming

Lagrangian Decomposition

- ▶ Integer and Combinatorial Optimization
- ▶ Lagrangian Relaxation

Lagrangian Function

The general mathematical-programming problem of minimizing $f(\boldsymbol{x})$ subject to a set of constraints $\{g_i(\boldsymbol{x}) \leq b_i\}$ has associated with it a Lagrangian function defined as $L(\boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + \sum_i \lambda_i[g_i(\boldsymbol{x}) - b_i]$, where the components λ_i of the nonnegative vector $\boldsymbol{\lambda}$ are called Lagrange multipliers. For a primal linear-programming problem, the Lagrange multipliers can be interpreted as the variables of the corresponding dual problem.

See

- ▶ Lagrangian Relaxation
- ▶ Nonlinear Programming

Lagrangian Relaxation

Monique Guignard
University of Pennsylvania, Philadelphia, PA, USA

Introduction

Many practical optimization problems include decision variables that are integer or 0-1. These problems, called mixed-integer programming problems or MIP for short, are in general difficult to solve, and there have been traditionally two classes of approaches to solve them: branch-and-bound or enumeration, and heuristic methods, either ad hoc or generic. Broadly speaking, branch-and-bound methods construct a tree, usually

S.I. Gass, M.C. Fu (eds.), *Encyclopedia of Operations Research and Management Science*,
DOI 10.1007/978-1-4419-1153-7,

binary, that allows the systematic exploration of all integer or 0-1 combinations of the discrete variables. Logical considerations and/or bounds on the optimal value computed as one moves down the tree may allow the pruning of a branch and backtracking to its root because one discovers that it would lead to infeasibilities or inferior solutions. Typically, bounds are obtained by solving a simpler, relaxed, optimization problem, most of the time the continuous relaxation of the MIP problem in which integer or 0-1 variables are allowed to take on fractional values. Heuristics, on the other hand, search for better and better feasible integer solutions, and do not usually compute bounds on the optimum, and therefore, even though they are getting more and more sophisticated and excel at finding optimal or near optimal solutions, cannot guarantee the quality of the solutions found.

Lagrangian relaxation stands somehow at the crossroads of both approaches. More powerful in terms of bound quality than the continuous relaxation, it also produces partially infeasible, but integer, solutions. These can usually serve as excellent starting points for specialized heuristics, referred to as Lagrangian heuristics. Contrary to the general heuristics mentioned above, given that one has found a bound called the Lagrangian bound, one knows whether the best solution found is good enough, or if it requires further investigation.

There is an enormous amount of literature devoted to the theory and applications of Lagrangian relaxation, starting with the seminal papers of Held and Karp (1970, 1971) and of Geoffrion (1974), although one could trace it back to earlier sources, for instance Everett's multipliers work (1963). Some early guides include (Fisher 1981, 1985).

Some of the questions to be addressed: Why use Lagrangian relaxation for integer programming problems? How does one construct a Lagrangian relaxation? What tools are there to analyze the strength of a Lagrangian relaxation? Are there more powerful extensions than standard Lagrangian relaxation, and when should they be used? Why is it that one can sometimes solve a strong Lagrangian relaxation by solving trivial subproblems? How does one compute the Lagrangian relaxation bound? Can one take advantage of Lagrangian problem decomposition? Does the strength of the model used make a difference in terms of bounds? Can one strengthen Lagrangian relaxation bounds by cuts, either kept or dualized? How can one design a Lagrangian heuristic? Can one achieve better results by remodeling the problem prior to doing Lagrangian relaxation?

The problems considered here have some integer variables, linear objective functions and constraints, and everything described below applies to maximization as well as minimization problems via the trivial sign transformations:

$$\text{Max } \{f(x)|x \in V\} = -\text{Min } \{-f(x)|x \in V\}.$$

Notation

If (P) is an optimization problem,

FS(P) denotes	the set of feasible solutions of problem (P)
OS(P)	the set of optimal solutions of problem (P)
v(P)	the optimal value of problem (P)
u^k, s^k, etc.,	the value of u, s, etc., used at iteration k
x^T	the transpose of x
x^k	the k^{th} extreme point of some polyhedron (see context)
$x^{(k)}$	a solution found at iteration k.
$\subset$	denotes strict inclusion.
Co(V)	denotes the convex hull of set V.

Relaxations of Optimization Problems

Geoffrion (1974) formally defines a relaxation of a generic minimization problem as follows.

Definition 1. *Problem* (RP_{min}): Min $\{g(x)|x \in W\}$ *is a relaxation of problem* (P_{min}): Min $\{f(x) \mid x \in V\}$ *if and only if* (i) *the feasible set of* (RP_{min}) *contains that of* (P_{min}), *and* (ii) *over the feasible set V of* (P_{min}), *the objective function of* (RP_{min}) *dominates* (*is better than*) *that of* (P_{min}), *i.e.*, $\forall\, x \in V$, $g(x) \leq f(x)$.

It clearly follows that $v(RP_{min}) \leq v(P_{min})$, in other words ($RP_{min}$) is an optimistic version of (P_{min}): it has more feasible solutions than (P_{min}), and for feasible solutions of (P_{min}), its own objective function is at least as good as (smaller than or equal to) that of (P_{min}), thus it has a smaller minimum.

Lagrangian Relaxation (LR)

In the rest of the note, (P) is assumed to be of the form $\text{Min}_x\{fx|Ax \leq b, Cx \leq d, x \in X\}$, where X contains the integrality restrictions on x, i.e. $X = \mathbb{R}^{n-p} \times \mathbb{Z}^p$, or $X = \mathbb{R}^{n-p} \times \{0,1\}^p$. Let I($X$) be the set of the p indices of x restricted to be integer (or binary). The constraints $Ax \leq b$ are assumed complicating, in the sense that, without them, problem (P) would be much simpler to solve. The constraints $Cx \leq d$ (possibly empty) will be kept, together with X, to form the Lagrangian relaxation of (P) as follows. Let λ be a nonnegative vector of weights, called Lagrangian multipliers.

Definition 2. *The Lagrangian relaxation of* (P) *relative to the complicating constraints* $Ax \leq b$, *with nonnegative Lagrangian multipliers* λ, *is the problem* (LR_λ) $\text{Min}_x\{f\ x + \lambda(Ax - b)|Cx \leq d, x \in X\}$.

Notice that (LR_λ) is still an integer programming problem, so its solutions, unlike those of the continuous relaxation, are integer solutions. However they need not be feasible solutions of (P), as they may violate some, or all, of the complicating constraints $Ax \leq b$, which are not enforced any more. In (LR_λ), the slacks of the complicating constraints $Ax \leq b$ have been added to the objective function with weights λ. One says that the constraints $Ax \leq b$ have been dualized. (LR_λ) is a relaxation of (P), since (i) FS (LR_λ) contains FS(P), and (ii) for any x feasible for (P), and any $\lambda \geq 0$, $fx + \lambda(Ax - b)$ is less than or equal to fx (i.e., not worse, since it is a minimization problem). It follows that $v(\text{LR}_\lambda) \leq v(\text{P})$, for all $\lambda \geq 0$, i.e., the optimal value $v(\text{LR}_\lambda)$, which depends on λ, is a lower bound on the optimal value of (P).

Definition 3. *The problem of finding the tightest Lagrangian lower bound on* v(P) *, i.e.*, (LR) $\text{Max}_{\lambda \geq 0}$ $v(\text{LR}_\lambda)$, *is called the Lagrangian dual of* (P) *relative to the complicating constraints* $Ax \leq b$. v(LR) *is called the Lagrangian relaxation bound, or simply the Lagrangian bound.*

Let (LP) denote the linear programming relaxation of problem (P). By LP duality, any Lagrangian relaxation bound is always at least as good as the LP bound, i.e., v(P), never worse. Notice also that (LR) is a problem in the dual space of the Lagrangian multipliers, whereas (LR_λ) is a problem in x, i.e., in the primal space.

Feasible Lagrangian solution

Let $x(\lambda)$ denote an optimal solution of (LR_λ) for some $\lambda \geq 0$, then $x(\lambda)$ is called a Lagrangian solution. One may be tempted to think that a Lagrangian solution $x(\lambda)$ that is feasible for the integer problem (i.e., that satisfies the dualized constraints) is also optimal for that problem. In fact this is generally not the case. What is true is that the optimal value of (P), v(P), lies in the interval between $fx(\lambda)+\lambda[Ax(\lambda)-b]$ and $fx(\lambda)$, where $fx(\lambda)$ is the value of a feasible solution of (P), thus an upper bound on v(P), and $fx(\lambda)+\lambda[Ax(\lambda)-b]$ is the optimal value of the Lagrangian problem (LR_λ), thus a lower bound on v(P). If, however, complementary slackness holds, i.e., if $\lambda[Ax(\lambda)-b]$ is 0, then $fx(\lambda)+\lambda[Ax(\lambda)-b] = v(\text{P}) = fx(\lambda)$, and $x(\lambda)$ is an optimal solution for (P).

Theorem 1. (1) *If* $x(\lambda)$ *is an optimal solution of* (LR_λ) *for some* $\lambda \geq 0$, *then* $fx(\lambda)+\lambda[Ax(\lambda)-b] \leq v(\text{P})$. *If in addition* $x(\lambda)$ *is feasible for* (P), *then* $fx(\lambda) +\lambda[Ax(\lambda)-b] \leq v(\text{P}) \leq fx(\lambda)$.

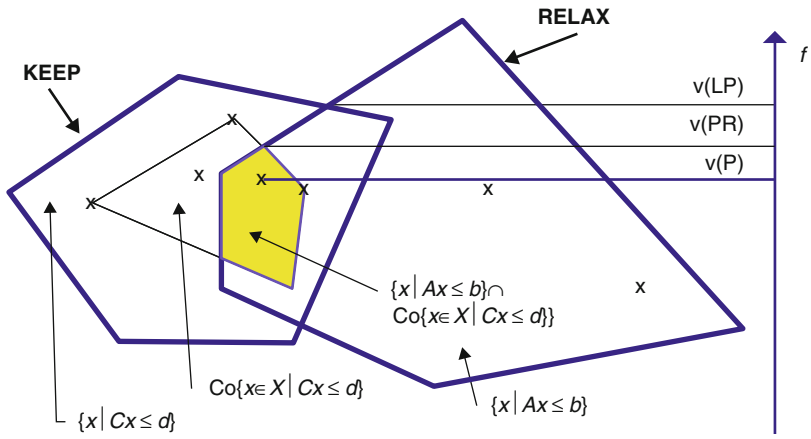
(2) *If in addition* $\lambda[Ax(\lambda)-b] = 0$, *then* $x(\lambda)$ *is an optimal solution of* (P), *and* $v(\text{P}) = fx(\lambda)$.

Remarks. Notice first that (2) is a sufficient condition of optimality, but it is not necessary. I.e., it is possible for a feasible $x(\lambda)$ to be optimal for (P), even though it does not satisfy complementary slackness. If the constraints that are dualized are equality constraints, and if $x(\lambda)$ is feasible for (P), complementary slackness holds automatically, thus $x(\lambda)$ is an optimal solution of (P), with $v(\text{P}) = fx(\lambda)$.

Geometric Interpretation

The following theorem, from (Geoffrion 1974), is probably what sheds most light on Lagrangian relaxation. It gives a geometric interpretation of the Lagrangian dual problem in the x- space, i.e., in the primal space, and this permits an in-depth study of the strength of specific Lagrangian relaxation schemes.

Theorem 2. *The Lagrangian dual* (LR) *is equivalent to the primal relaxation* (PR) $\text{Min}_x\{fx|Ax \leq b, x \in Co\{x \in X|Cx \leq d\}\}$, *in the sense that* v(LR) = v(PR) (Fig. 1).



Lagrangian Relaxation, Fig. 1 Geometric interpretation of Lagrangian relaxation

This result is based on LP duality and properties of optimal solutions of linear programs. Remember though that this result may not be true if the constraint matrices are not rational.

The following important definition and results follow from this geometric interpretation.

Definition 4. *One says that* (LR) *has the Integrality Property* (IP *for short*) *if* $\mathrm{Co}\{x \in X | Cx \leq d\} = \{x \in \mathbb{R}^n | Cx \leq d\}$.

If (LR) has the Integrality Property, then the extreme points of $\{x \in \mathbb{R}^n | Cx \leq d\}$ are in X. The unfortunate consequence of this property, as stated in the following corollaries, is that such an LR scheme cannot produce a bound stronger than the LP bound. Sometimes, however, this is useful anyway because the LP relaxation cannot be computed easily. This may be the case for instance for some problems with an exponential number of constraints that can be relaxed anyway into easy to solve subproblems. The traveling salesman problem is an instance of a problem which contains an exponential number of (subtour elimination) constraints. A judicious choice of dualized constraints leads to Lagrangian subproblems that are 1-tree problems, thus eliminating the need to explicitly write all the subtour elimination constraints (Held and Karp 1970, 1971).

Here are the two corollaries of Theorem 2 that explain the important role played by the Integrality Property.

Corollary 1. *If* $\mathrm{Co}\{x \in X | Cx \leq d\} = \{x \in \mathbb{R}^n | Cx \leq d\}$, *then* $v(\mathrm{LP}) = v(\mathrm{PR}) = v(\mathrm{LR}) \leq v(\mathrm{P})$.

In that case, the Lagrangian relaxation bound is *equal* to (cannot be better than) the LP bound.

Corollary 2. *If* $\mathrm{Co}\{x \in X | Cx \leq d\} \subset \{x \in \mathbb{R}^n | Cx \leq d\}$, *then* $v(\mathrm{LP}) \leq v(\mathrm{PR}) = v(\mathrm{LR}) \leq v(\mathrm{P})$, *and it may happen that the Lagrangian relaxation bound is strictly better than the LP bound.*

Unless (LR) does *not* have the Integrality Property, it will *not* yield a stronger bound than the LP relaxation. It is thus important to know if all vertices of the rational polyhedron $\{x \in \mathbb{R}^n | Cx \leq d\}$ are in X.

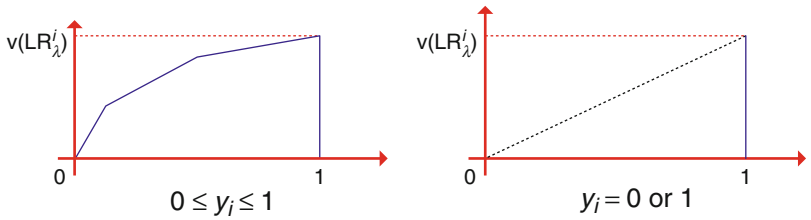
Easy-to-Solve Lagrangian Subproblems

It may happen that Lagrangian subproblems, even though in principle hard to solve because they do not have the Integrality Property, are in fact much easier to solve through some partial decomposition; they can sometimes even be solved in polynomial time, by exploiting their special structure. It is of course important to be able to recognize such favorable situations, especially if one can avoid using Branch-and-Bound to solve them. It should be noted that these favorable cases do not in general occur naturally, but only after some constraint(s) have been dualized, due to a weakening of the original links between continuous and integer variables.

One case is due to what is sometimes called the Integer Linearization Property (or ILP for short) for mixed 0-1 problems.

Integer Linearization Property

Geoffrion (1974) and Geoffrion and McBride (1978) described and used this important property of *some* Lagrangian subproblems. W.l.o.g., assume that all



Lagrangian Relaxation, Fig. 2 Integer linearization property

variables are indexed by $i \in I$, and maybe by some additional indices, and that some of the 0-1 variables are called y_i. If, except for constraints containing only these 0-1 variables y_i, the Lagrangian problem, say, (LR_λ), has the property that the value taken by a given y_i decides alone the fate of all other variables containing the same value of the index i – that usually means that if variable y_i is 0, all variables in its family are 0, and if it is 1, they are solutions of a subproblem – one may be able to reformulate the problem in terms of the variables y_i only. Often, but not always, when this property holds, it is because the Lagrangian problem, after removal of all constraints containing only the y_i's – call it (LRP_λ), for partial problem – decomposes into one problem (LRP^i_λ) for each i, i.e., for each 0-1 variable y_i. The use of this property is based on the following fact. In problem (LRP^i_λ), the integer variable y_i can be viewed as a parameter, however one does know that for the mixed-integer problem (LRP^i_λ), the feasible values of that parameter are only 0 and 1, and one can make use of the fact that there are only two possible values for $v(LRP^i_\lambda)$, the value computed for y_i=1, say v_i $(= v_i.y_i$ for y_i=1), and the value for y_i=0, that is, 0 $(= v_i.y_i$ for y_i=0), which implies that for all possible values of y_i, $v(LRP^i_\lambda) = v_i.y_i$. Hence the name integer linearization, as one replaces a piecewise linear function corresponding to $0 \le y_i \le 1$ by a line through the points (0, 0) and (1, v_i) (Fig. 2).

One may in such cases obtain LR bounds much tighter than the LP bounds, even though the subproblems are trivial to solve.

Constructing a Lagrangian Relaxation

There are often many ways in which a given problem can be relaxed in a Lagrangian fashion. A few standard ones are listed here, mostly to point out that often some reformulation prior to relaxation can help, and that for many complex models, intuition and some understanding of the constraint interactions may suggest ingenious and efficient relaxation schemes.

(1) **One can isolate an interesting subproblem and dualize the other constraints**.

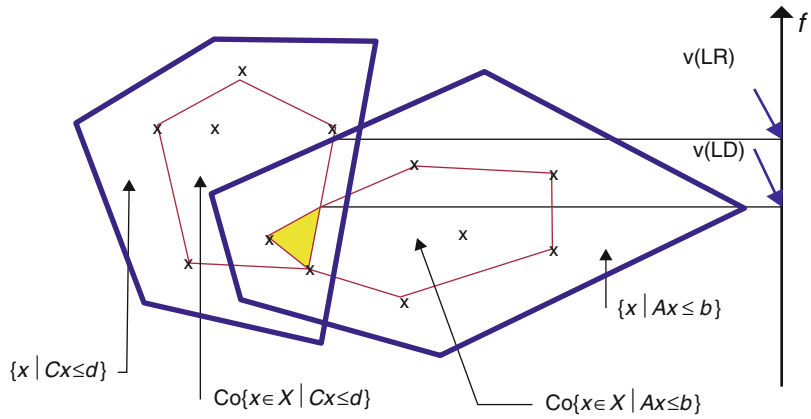
This is the most commonly used approach. It has the advantage that the Lagrangian subproblems are interesting (in the sense usually that they have a special structure that can be exploited) and there may even exist specialized algorithms for solving them efficiently.

(2) **If there are two (or more) interesting subproblems with common variables, one can split these variables first, then dualize the copy constraint.**

This is called Lagrangian decomposition (LD) (Soenen 1977), variable splitting (Näsberg et al. 1985), or variable layering (Glover and Klingman 1988). One must first reformulate the problem using variable splitting, in other words, one must rename the variables in part of the constraints as if they were independent variables. Problem (P): $\mathrm{Min}_x\{f\ x | Ax \le b, Cx \le d, x \in X\}$ is clearly equivalent to problem (P′): $\mathrm{Min}_{x,y}\{f\ x | Ax \le b, x \in X, Cy \le d, y \in X, x = y\}$, in the sense that they have equal optimal values (but notice that they have different variable spaces). In addition if x^* is an optimal solution of (P), then the solution $(x, y) \equiv (x^*, x^*)$ is optimal for (P′), and if (x^*, y^*) is an optimal solution of (P′) with $x^* = y^*$, then x^* is optimal for (P). One dualizes the copy constraint $x = y$ in (P′) with multipliers λ, this separates the problem into an x-problem and a y-problem: (LD_λ) $\mathrm{Min}_{x,y}\{fx + \lambda(y - x) | Ax \le b, x \in X,\ Cy \le d,\ y \in X\} = \mathrm{Min}_x\{(f - \lambda)x \mid Ax \le b,\ x \in X\} + \mathrm{Min}_y\{\lambda y \mid Cy \le d, y \in X\}$.

This process creates a staircase structure, and thus decomposability, in the model. Notice that here λ is not required to be nonnegative.

Remember also that when one dualizes equality constraints, a feasible Lagrangian solution is



Lagrangian Relaxation, Fig. 3 Geometric interpretation of Lagrangean decomposition

automatically optimal for the original integer programming problem. The copy constraint being an equality constraint, if both Lagrangian subproblems have the same optimal solution, that solution is optimal for the IP problem.

Guignard and Kim (1987) showed that the LD bound can strictly dominate the LR bounds obtained by dualizing either set of constraints:

Theorem 3.

$$\begin{aligned} &\textit{If } v(\mathrm{LD}) = \mathrm{Max}_{\lambda}[\mathrm{Min}_{x}\{(f-\lambda)x|\ Ax \le b, x \in X\} \\ &\qquad + \mathrm{Min}\{\lambda y|Cy\ \le\ d, y \in X\}] \textit{ then} \\ &\qquad v(\mathrm{LD}) = \mathrm{Min}\ \{fx^{y}|\, x \in \mathrm{Co}\{x \in X|Ax \le b\} \\ &\qquad \cap\, \mathrm{Co}\{x \in X|Cx \le d\}. \end{aligned}$$

This new geometric interpretation is demonstrated in Fig. 3.

Corollary 3.

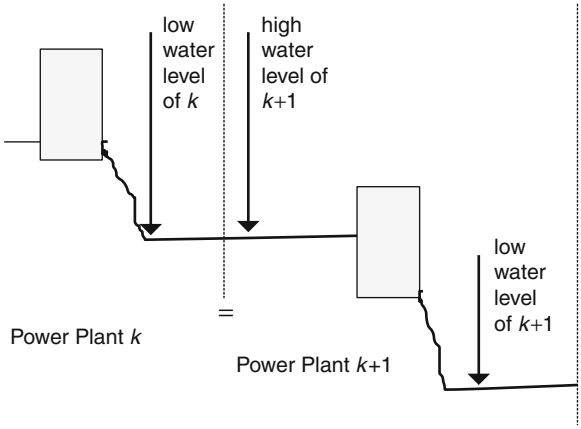
- *If one of the subproblems has the Integrality Property, then v(LD) is equal to the better of the two LR bounds corresponding to dualizing either* $Ax \le b$ *or* $Cx \le d$.
- *If both subproblems have the Integrality Property, then* $v(LD) = v(LP)$.

A very important application of the splitting variable scheme can be found in stochastic optimization, when the uncertainty is represented by 2-stage or multistage scenario trees. The non-anticipativity constraints (or NAC) must be satisfied by the variables attached to the scenario groups or nodes in the tree. Splitting variables in the NAC and dualizing the copy constraints produces a Lagrangean decomposition of the Deterministic Equivalent Model. See Escudero (2009) and Birge and Louveaux (2011), among others.

Occasionally the variable splitting will correspond to a physical split of one of the problem's decision variables. This is illustrated by the following example.

Example 1. Guignard and Yan (1993) described the following problem and scheme for a hydroelectric power management problem.

Electric utility production planning is the selection of power generation and energy efficiency resources to meet customer demands for electricity over a multi-period time horizon. The project described in the paper is a real-world hydropower plant operations management problem of a dispatch type. The system consists of a chain of 10 consecutive hydropower plants separated by reservoirs and falls with 23 identical machines installed to generate electric power. Specifically there are two machines installed in eight power plants (plants 1, 2, 3, 4, 5, 6, 7, and 10), three machines in one power plant (plant 8) and four machines in the last power plant (plant 9). Each machine has two or four work parts for producing electric power, according to different water throughput. Since demand for electric power varies with different time periods, power plant managers must make optimal decisions concerning the number of machines that should be operated in each power plant during each time period. Managing the power generation requires decisions concerning water



Lagrangian Relaxation, Fig. 4 Lagrangian decomposition splits the water level

releases at each plant k in each time period. A period is two hours. The model (which is confidential) was constructed by an independent consulting firm. This results in a large mixed-integer program. The problem is complex, with 2,691 variables, 384 of which are binary, and 12,073 constraints. The firm had tried to solve the problem for the utility company with several of the best MIP software packages available, with help from the software companies themselves. Yet they did not succeed. Guignard and Yan repeated the tests with several solvers running under GAMS, on several RISC systems, also to no avail. The best result after 5 days and six hours on an HP workstation was a bracket [3174.97, 3534.17], i.e., a residual gap of more than 11%.

In order to reduce the complexity of the model, they tried several Lagrangian relaxations and decompositions. One of the decompositions tested consists in "cutting" each reservoir in half (see Fig. 4), i.e. splitting the water level variable in each reservoir, and dualizing the following copy constraint:

$$\text{high water level in } k+1 = \text{low water level in } k.$$

This Lagrangian decomposition produces one power management problem per power plant k. These subproblems do not have a special structure, but are much simpler and smaller than the original problem, are readily solvable by commercial software, and do not have the Integrality Property. They were solved by Branch-and-Bound.

This LD shrinks problem size, and yields Lagrangian bounds much stronger than the LP bounds. In addition the Lagrangian solutions can be modified to provide feasible schedules.

(3) **One can dualize linking constraints**:

After possibly some reformulation, problems may contain independent structures linked by some constraints: $\text{Min}_{x,y}\{f\ x+gy|Ax \le b,\ x \in X,\ Cy \le d,\ y \in Y,\ Ex+Fy \le h\}$. Dualizing the linking constraints $Ex + Fy \le h$ splits the problem into an x-problem and a y-problem. The original problem may only contain x and some reformulation introduces a new variable y, while the relationship between x and y is captured by the new constraints $Ex + Fy \le h$.

Example 2. A production problem over multiple facilities contains constraints related to individual facilities, while the demand constraints link all plant productions. If one dualizes the demand constraints, the Lagrangian problem decomposes into a production problem for each facility, which is typically much easier to solve than the overall problem. If at least one of these subproblems does not have the Integrality Property, this LR may yield a tighter bound than the LP bound. In (Andalaft et al. 2003), a forest company must harvest geographically distinct areas, and dualizing the demand constraints splits the problem into one subproblem per area, which is typically much easier to solve than the overall problem.

(4) **One can sometimes dualize aggregate rather than individual copies of variables.**

Instead of creating a copy y of variable x and introducing y into model (P) by rewriting the constraint $Cx \le d$ as $Cy \le d$, to yield the equivalent model (P′): $\text{Min}_{x,y}\{f\ x|Ax \le b,\ x \in X,\ Cy \le d,\ y \in X,\ x = y\}$, one can also create a problem (P″) equivalent to problem (P) by introducing a new variable y and forcing the constraint $Dy = Cx$. This constraint is in general weaker than the constraint $x = y$. Model (P″) is $\text{Min}_{x,y}\{f\ x|Ax \le b,\ x \in X,\ Dy \le d,\ y \in X,\ Dx = Cy\}$. The LR introduced here dualizes the aggregate copy constraint $Dx = Cy$.

Notice that the copy constraint is an equality constraint, therefore if the Lagrangian subproblems have optimal solutions x and y that satisfy the aggregate copy constraint, i.e., if $Dy = Cx$, then the x- solution is optimal for the IP problem.

Example 3. Consider the bi-knapsack problem

(BKP)$\text{Max}_x\{\sum_i c_i x_i | \sum_i b_i x_i \le m,\ \sum_i d_i x_i \le n, x_i \in \{0,1\}, \forall i\}$.

One can introduce a new variable y, and write $\sum_i b_i x_i = \sum_i b_i y_i$. The equivalent problem is

(BKP')$\text{Max}_{x,y}\{\sum_i c_i x_i | \sum_i b_i y_i \le m,\ \sum_i d_i x_i \le n, \sum_i b_i x_i = \sum_i b_i y_i,\ x_i, y_i \in \{0,1\}, \forall i\}$

and the LR problem is

$$(\text{LR}_\lambda)\text{Max}_{x,y}\left\{\sum_i c_i x_i - \lambda\left(\sum_i b_i x_i - \sum_i b_i y_i\right)\right| \sum_i b_i y_i \le m, \sum_i d_i x_i \le n,\ x_i, y_i \in \{0,1\}, \forall i\}$$
$$= \text{Max}_x\left\{\sum_i (c_i - \lambda b_i) x_i \middle| \sum_i d_i x_i \le n, x_i \in \{0,1\}, \forall i\right\} + \text{Max}_y\left\{\lambda \sum_i b_i y_i) \middle| \sum_i b_i y_i \le m,\ y_i \in \{0,1\}, \forall i\right\}.$$

Here λ is a single real multiplier of arbitrary sign. The Lagrangian bound produced by this scheme is in between that of the LP bound and that of the Lagrangian decomposition bound obtained by dualizing $x_i = y_i\ \forall i$. This is similar in spirit to the copy constraints introduced in Reinoso and Maculan (1992).

It would seem natural that a reduction in the number of multipliers should imply a reduction in the quality of the LR bound obtained. This is not always the case, however, as shown in example 4.

Example 4. Chen and Guignard (1998) considered an aggregate Lagrangian relaxation of the capacitated facility location problem. The model uses continuous variables x_{ij} that represent the percentage of the demand d_j of customer j supplied by facility i, and binary variables y_i, equal to 1 if facility i with capacity a_i is operational. The constraint $\sum_j d_j x_{ij} \le a_i y_i$ imposes a conditional capacity restriction on the total amount that can be shipped from potential facility i.

(CPLP)

$\text{Min}_{x,y}\ \Sigma_i \Sigma_j c_{ij} x_{ij} + \Sigma_i f_i y_i$		
s.t. $\Sigma_i x_{ij} = 1$, all j	(D)	*meet 100% of customer demand*
$x_{ij} \le y_i$, all i, j	(B)	*ship nothing if plant is closed*
$\Sigma_i a_i y_i \ge \Sigma_j d_j$,	(T)	*enough plants to meet total demand*
$\Sigma_j d_j x_{ij} \le a_i y_i$, all i	(C)	*ship no more than plant capacity*
$x_{ij} \ge 0$, $y_i = 0$ or 1, all i, j.		

Constraint (T) is redundant, but may help getting tighter Lagrangian relaxation bounds.

The three best Lagrangian schemes are:

(LR) (Geoffrion and McBride 1978)

One dualizes (D) then uses the integer linearization property. The subproblems to solve are one continuous knapsack problem per plant ((C) with $y_i = 1$) and one 0-1 knapsack problem over all plants (constraint (T)). The Lagrangian relaxation bound is tight, and it is obtained at a small computational cost.

(LD) (Guignard and Kim 1987).

Duplicate (T). Make copies $x_{ij} = x'_{ij}$ and $y_y = y'_i$ and use x'_{ij} and y'_i in (C) and in one of the (T)'s. One obtains the split

{(D), (B), (T)} → APLP

{(B), (T), (C)} → this is like in (LR)

This LD bound is tighter than the (LR) bound, but expensive to compute, in particular because of a large number of multipliers.

(LS) (Chen and Guignard 1998).

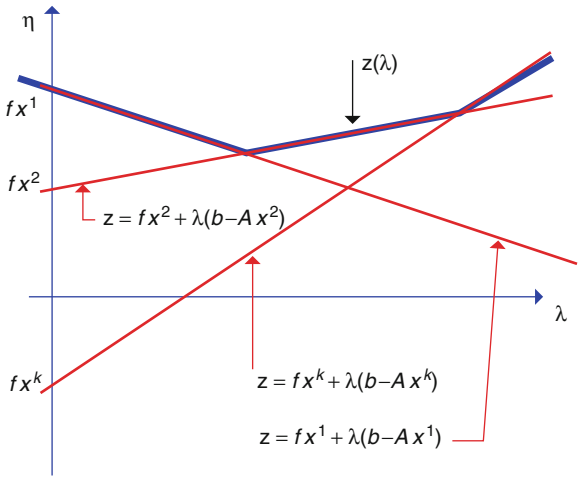
Copy $\sum_j d_j x_{ij} = \sum_j d_j x'_{ij}$ and $y_i = y'_i$ in (C). This yields the same split as (LD), and the same bound. This is very surprising, as it is less expensive to solve (LS) than (LD), in particular because (LS) has far fewer multipliers.

In example 4, creating new copy variables x'_{ij} and y'_i, one can create an LS by dualizing the aggregate (linking) copy constraints $\sum_j d_j x_{ij} = \sum_j d_j x'_{ij}$ and $a_i y_i = a_i y'_i$. Surprisingly, one can prove that the LS bound for this problem is as strong as the LD bound obtained by dualizing individual copies $x_{ij} = x'_{ij}$ and $y_i = y'_i$. This suggests that "aggregating" variables before copying them may be an attractive alternative to Lagrangian decomposition, at least for some problem structures. A more general structure than CPLP is actually described in Chen and Guignard (1998).

Characteristics of the Lagrangian Function

The Lagrangian function $z(\lambda) = v(\text{LR}_\lambda)$ is an implicit function of λ. Suppose that the set $\text{Co}\{x \in X | Cx \le d\}$ is a polytope, i.e., a bounded polyhedron, then there exists a finite family $\{x^1, x^2, \ldots, x^K\}$ of extreme points of $\text{Co}\{x \in X | Cx \le d\}$, i.e., of points of $\{x \in X | Cx \le d\}$, such that $\text{Co}\{x \in X | Cx \le d\} = \text{Co}\ \{x^1, x^2, \ldots, x^K\}$. It then follows that

$$\text{Min}_x\{fx + \lambda(b - Ax) | Cx \le d, x \in X\} = \text{Min}_{k=1,\ldots,K}\{f\ x^k + \lambda(b - Ax^k)\}$$



Lagrangian Relaxation, Fig. 5 The Lagrangean function of a maximization problem

and $z(\lambda)$ is the lower envelope of a family of linear functions of $\lambda, f\ x^k + \lambda\left(b - Ax^k\right), k=1,\ldots,K$, and thus is a concave function of λ, with breakpoints where it is not differentiable, i.e., where the optimal solution of (LR_λ) is not unique. Figure 5 shows a Lagrangian function for the case where (P) is a maximization problem, this (LR) is a minimization problem, and $z(\lambda)$ a convex function of (λ).

A concave function $f(x)$ is continuous over the relative interior of its domain, and it is differentiable almost everywhere, i.e., except over a set of measure 0. At points where it is not differentiable, the function does not have a gradient, but is always has subgradients.

Definition 5. *A vector* $y \in (\mathrm{R}^n)^*$ is *a* subgradient *of a concave function* $f(x)$ *at a point* $x^0 \in \mathrm{R}^n$ *if for all* $x \in \mathrm{R}^n$

$$f(x) - f\left(x^0\right) \leq y \cdot (x - x^0).$$

Definition 6. *The set of all subgradients of a concave function* $f(x)$ *at a point* x^0 *is called the subdifferential of* f *at* x^0 *and it is denoted* $\partial f(x^0)$.

Theorem 4. *The subdifferential* $\partial f(x^0)$ *of a concave function* $f(x)$ *at a point* x^0 *is always nonempty, closed, convex and bounded.*

If the subdifferential of f at x^0 consists of a single element, that element is the gradient of f at x^0, denoted by $\nabla f(x^0)$.

The dual problem (LR) is

$$\begin{aligned}&\mathrm{Max}_{\lambda\ \geq 0} v(\mathrm{LR}_\lambda) = \mathrm{Max}_{\lambda \geq 0} z(\lambda) = \\ (\mathrm{LR})\ &\mathrm{Max}_{\lambda\ \geq 0} \mathrm{Min}_{k=1,\ldots,K}\left\{fx^k + \lambda\left(b - Ax^k\right)\right\} = \\ &\mathrm{Max}_{\lambda\ \geq 0,\eta}\left\{\eta | \eta \leq fx^k + \lambda(b - Ax^k), k = 1,\ldots,K\right\}.\end{aligned}$$

Let λ^* be a minimizer of $z(\lambda)$, $\eta^* = z(\lambda^*)$, λ^k be a current "guess" at λ^*, let $\eta_k = z(\lambda^k)$, and $\mathrm{H}_k = \left\{\lambda | fx^k + \lambda(b - Ax^k) = \eta^k\right\}$ be a level hyperplane passing through λ^k.

- If $z(\lambda)$ is differentiable at λ^k, i.e., if (LR_λ) has a unique optimal solution x^k, it has a **gradient** $\nabla z(\lambda^k)$ at λ^k:

$$\nabla^{\mathrm{T}} z(\lambda^k) = (b - Ax^k) \perp \mathrm{H}_k.$$

- If $z(\lambda)$ is nondifferentiable at λ^k, i.e., if (LR_λ^k) has multiple optimal solutions, the vector $s^k = (b - Ax^k)^{\mathrm{T}}$ is a subgradient of $z(\lambda)$ at λ^k. That vector s^k is orthogonal to H_k.

If one considers the contours $\mathrm{C}\{k\} = \left\{\lambda \in \mathbb{R}_+^m | z(\lambda) \geq a\right\}$, α a scalar, these contours are convex polyhedral sets. See Fig. 6.

Note: A subgradient is not necessarily a direction of increase for the function, even locally, as seen on Fig. 6.

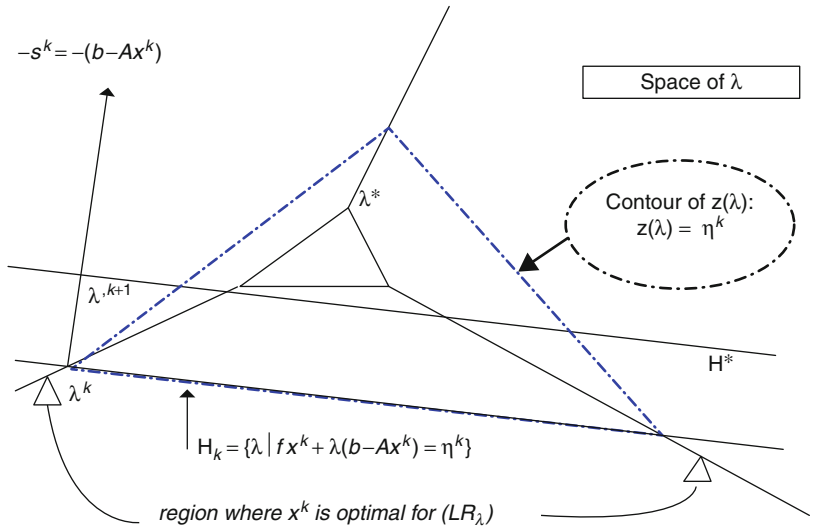
Theorem 5. *The vector* $(b - Ax^k)^{\mathrm{T}}$ is a *subgradient* of $z(\lambda)$ at λ^k.

Primal and Dual Methods to Solve Relaxation Duals

A number of methods have been proposed to solve Lagrangian duals. They are either ad-hoc, like for instance dual ascent methods, or general purpose, usually aiming at solving a generic nonsmooth convex optimization problem. This section reviews the most important approaches.

Subgradient Method

This method was proposed in (Held and Karp 1971). It is an iterative method in which at iteration k, given the current multiplier vector λ^k, a step is taken along a subgradient of $z(\lambda^k)$, then, if necessary, the resulting point is projected onto the nonnegative orthant.



Lagrangian Relaxation, Fig. 6 Contours and subgradient

Let $x^{(k)}$ be an optimal solution of (LR^k_λ). Then $s^k = (b - Ax^{(k)})^T$ is a subgradient of z(λ) at λ^k. If λ^* is an (unknown) optimal solution of (LR), with $\eta^* = z(\lambda^*)$, let λ'^{k+1} be the projection of λ^k on the hyperplane H^* parallel to H_k, defined by

$$H^* = \left\{\lambda | fx^k + \lambda(b - Ax^{(k)}) = \eta^*\right\}.$$

The vector s^k is perpendicular to both H_k and H^*, therefore $\lambda'^{k+1} - \lambda^k$ is a nonnegative multiple of s^k:

$$\lambda'^{k+1} - \lambda^k = \mu s^k, \mu \geq 0.$$

Also, λ'^{k+1} belongs to H^*:

$$f\ x^{(k)} + \lambda'^{k+1}(b - Ax^{(k)}) = \eta^*,$$

therefore $f\ x^k + \mu s^k(b - Ax^{(k)}) = \eta^k + \mu s^k.s^k = \eta^*$
and $\mu = (\eta^* - \eta^k)/||s^k||^2$,
so that $\lambda'^{k+1} = \lambda^k + s^k.\ (\eta^* - \eta^k)\ /||s^k||^2$.

Finally define $\lambda^{k+1} = [\lambda'^{k+1}]^+$, i.e., define the next iterate λ^{k+1} as the projection of λ'^{k+1} onto the nonnegative orthant, as λ must be nonnegative. Given the geometric projections described above, it is clear that λ^{k+1} is closer to λ^* than λ^k, thus the sequence $||\lambda^k - \lambda^*||^2$ is monotone nonincreasing.

Remark. This formula unfortunately uses the unknown optimal value η* of (LR). One can try to use an estimate for that value, but then one may be using either too small or too large a multiple of s^k. If one sees that the objective function values do not improve for too many iterations, one should suspect that η^* has been overestimated (for a maximization problem) and that one is overshooting, thus one should try to reduce the difference η^*-η^k. This can be achieved by introducing from the start a positive factor $\varepsilon_k \in (0,2)$, in the subgradient formula:

$$\lambda^{k+1} = \lambda^k + s^k.\ \varepsilon_k(\eta^* - \eta^k)\ /||s^k||^2,$$

and reducing the scalar ε_k when there is no improvement for too long.

Practical convergence of the subgradient method is unpredictable, sometimes quick and fairly reliable, sometimes erratic. Many authors have studied this problem and have proposed a variety of remedies.

Dual Ascent Methods

In this kind of approach, one takes advantage of the structure of the Lagrangian dual to create a sequence of multipliers that guarantee a monotone increase in Lagrangian function value. This approach had been pioneered by Bilde and Krarup (1967, 1977) for solving approximately the LP relaxation of the uncapacitated facility location problem (UFLP). General principles for developing a successful Lagrangian dual ascent method can be found in (Guignard and Rosenwein 1989).

Constraint Generation Method (Also Called Cutting Plane Method, or CP)

In this method , one uses the fact that $z(\lambda)$ is the lower envelope of a family of linear functions:

$$\begin{aligned}&\mathrm{Max}_{\lambda\geq 0} v(\mathrm{LR}_\lambda) = \mathrm{Max}_{\lambda\geq 0} z(\lambda) =\\ (\mathrm{LR})\ &\mathrm{Max}_{\lambda\geq 0}\mathrm{Min}_{k=1,\ldots,K}\left\{f\,x^k + \lambda\left(b - Ax^k\right)\right\} =\\ &\mathrm{Max}_{\lambda\geq 0,\eta}\left\{\eta \mid \eta \leq f\,x^k + \lambda\left(b - Ax^k\right), k = 1,\ldots,K\right\}.\end{aligned}$$

At each iteration k, one generates one or more cuts of the form

$$\eta \leq fx^k + \lambda(b - Ax^{(k)}),$$

by solving the Lagrangian subproblem (LR_λ^k) with solution $x^{(k)}$. These cuts are added to those generated in previous iterations to form the current LP master problem:

$$\left(\mathrm{MP}^k\right)\quad \mathrm{Max}_{\lambda\geq 0,\eta}\left\{\eta \mid \eta \leq fx^{(h)} + \lambda(b - Ax^{(h)}), h = 1,\ldots,k\right\},$$

whose solution is the next iterate λ^{k+1}. The process terminates when $v(\mathrm{MP}^k) = z(\lambda^{k+1})$. This value is the optimal value of (LR).

Column Generation (CG)

(CG) has been used extensively, in particular for solving very large scheduling problems (airline, buses, etc.). It consists in reformulating a problem as an LP (or an IP) whose activities (or columns) correspond to feasible solutions of a subset of the problem constraints, subject to the remaining constraints. The variables are weights attached to these solutions.

There are two aspects to column generation: first, the process is dual to Lagrangian relaxation and to CP. Secondly, it can be viewed as an application of Dantzig and Wolfe's decomposition algorithm (Dantzig and Wolfe 1960, 1961).

Let the $x^k \in \left\{x \in X \mid Cx^k \leq d\right\}$, $k \in K$, be chosen such that $\mathrm{Co}\left\{x^k\right\} = \mathrm{Co}\{x \in X \mid Cx \leq d\}$. A possible choice for the x^k's is all the points of $\mathrm{Co}\{x \in X \mid Cx \leq d\}$ but a cheaper option is all extreme points of $\mathrm{Co}\{x \in X \mid Cx \leq d\}$.

Problem (P): $\mathrm{Min}_x\{fx \mid Ax \leq b, Cx \leq d, x \in X\}$ yields the Lagrangian dual (i.e., in the λ-space) problem

$$(\mathrm{LR})\quad \mathrm{Max}_{\lambda\geq 0}\mathrm{Min}_x\{fx + \lambda(Ax - b) \mid Cx \leq d, x \in X\}$$

which is equivalent to the primal (i.e., in the x-space) problem

$$(\mathrm{PR})\quad \mathrm{Min}_x\{fx \mid Ax \leq b, x \in \mathbf{Co}\{x \in X \mid Cx \leq d\}\},$$

which itself can be rewritten as (PR) $\mathrm{Min}_x\left\{f\left(\sum_{k\in K}\mu_k x^k\right)\middle| A\left(\sum_{k\in K}\mu_k x^k\right)x \leq b\right\}$ given $= \mathrm{Min}_x\left\{\sum_{k\in K}\mu_k.(fx^k)\middle|\sum_{k\in K}\mu_k.(Ax^k) \leq b\right\}$, that one can write $x \in \mathrm{Co}\{x \in X \mid Cx \leq d\}$ as $x = \sum_{k\in K}\mu_k x^k$, with $\sum_{k\in K}\mu_k = 1$ and $\mu_k \geq 0$.

The separation of a problem into a master- and a sub-problem is equivalent to the separation of the constraints into kept and dualized constraints. The columns generated are solutions of integer subproblems that have the same constraints as the Lagrangian subproblems.

The value of the LP relaxation of the master problem is equal to the Lagrangian relaxation bound. The strength of a CG or LR scheme would then seem to be based on the fact that the subproblems do not have the integrality property. It may happen however that such a scheme can be successful at solving problems with the integrality property because it permits the indirect computation of v(LP) when this value could not be computed directly, e.g., because of an exponential number of constraints (Held and Karp 1970, 1971).

One substantial advantage of (CP) or (CG) over subgradient algorithms is the existence of a true termination criterion $v(\mathrm{MP}^k) = z(\lambda^{k+1})$.

Bundle Methods

Lemaréchal (1974) introduced an extension of subgradient methods, called bundle methods, in which past information is collected to provide a better approximation of the Lagrangian function. The standard CP algorithm uses the bundle of the subgradients that were already generated, and constructs a piecewise linear approximation of the Lagrangian function. This method is usually slow and unstable. Three different stabilization approaches have been proposed. At any moment, one has a model representing the Lagrangian function, and a so-called stability center, which should be a reasonable approximation of the true optimal solution.

One generates a next iterate which is a compromise between improving the objective function and keeping close to the stability center. The next iterate becomes the new stability center (a serious step) only if the objective function improvement is "good enough". Otherwise, one has a null step, in which however one improves the function approximation. In addition, this next iterate shouldn't be too far away from the stability center. The three stabilization approaches propose different ways of controlling the amount of move that is allowed. Either the next iterate must remain within a so-called trust region, or one adds a penalty term to the approximation of the function that increases with the distance from the stability center, or one remains within a region where the approximation of the function stays above a certain level (for a maximizaton problem). This proximity measure is the one parameter that may be delicate to adjust in practical implementations. There is a trade-off between the safety net provided by this small move concept, and the possibly small size of the bound improvement.

The Volume Algorithm (VA)

The volume algorithm (Barahona and Anbil 2000), an extension of the subgradient algorithm, can be seen as a fast way to approximate Dantzig-Wolfe decomposition, with a better stopping criterion, and it produces primal as well as dual vectors by estimating the volume below the faces that are active at an optimal dual solution. It has been used successfully to solve large-scale LP's arising in combinatorial optimization, such as set partitioning or location problems.

Subproblem Decomposition

In many cases, the Lagrangian subproblem decomposes into smaller problems, and this means that the feasible region is actually the Cartesian product of several smaller regions. One clear advantage is the reduction in computational complexity for the Lagrangian subproblems: indeed, it is generally much easier to solve 50 problems with 100 binary variables each, say, than a single problem with 5,000 (i.e., 50x100) binary variables.

It also means that in column generation, the columns (i.e., the vectors that are feasible solutions of the kept constraints) decompose into smaller subcolumns, and each subcolumn is a convex combination of extreme points of a small region. By assigning different sets of weights to these convex combinations, one allows mix-and-match solutions, in other words, one may combine a subcolumn for the first subproblem that was generated at iteration 10, say, with a subcolumn for the second subproblem generated at iteration 7, etc. , to form a full size column. If one had not decomposed the problem ahead of time, one may have had to wait a long time for such a complete column to be generated.

By duality, this means that in a cutting plane environment, one can also generate sub-cuts for each subproblem, which amounts to first replacing η by $z - \lambda b$ in

$$\left(\mathrm{MP}^k\right) \mathrm{Max}_{\lambda \geq 0, \eta}\left\{\eta | \eta \leq f\, x^{(h)} + \lambda(b - Ax^{(h)}), h = 1, \ldots, k\right\}$$
$$= \mathrm{Max}_{\lambda \geq 0, z}\left\{z - \lambda b | z \leq (f - \lambda A)x^{(h)}, h = 1, \ldots, k\right\},$$

and then z by a sum of scalars z_l, with $z_l \leq (f^l - \lambda A_l)\, x_l^{(h)}$, where l is the index of the Lagrangian subproblem, f^l, A_l, and $x_l^{(h)}$ are the l^{th} portions of the corresponding submatrices and vectors, and x_l^h is a Lagrangian solution of the l^{th} subproblem found at iteration h, yielding the disaggregated master problem

$$\left(\mathrm{MPD}^k\right) \mathrm{Max}_{\lambda \geq 0, z_l}\left\{\sum\nolimits_l z_l - \lambda b | z_l \leq (f - \lambda A)^l x_l^h, h = 1, \ldots, k\right\}.$$

Example 5. Consider the Generalized Assignment Problem, or GAP (for the minimization case, although it would work in exactly the same way with maximization).

$$\begin{aligned} (\mathrm{GAP})\ \mathrm{Min}\ & \sum\nolimits_i \sum\nolimits_j c_{ij} x_{ij} \\ \text{s.t.}\quad & \sum\nolimits_j a_{ij} x_{ij} \leq b_i, \quad \forall i \in I \quad (\mathrm{KP}) \\ & \sum\nolimits_i x_{ij} = 1, \quad \forall j \in J \quad (\mathrm{MC}) \\ & x_{ij} \in \{0, 1\}, \quad \forall i \in I, j \in J. \end{aligned}$$

Its strong Lagrangian relaxation is

$$(\mathrm{LR}_\lambda) \mathrm{Min} \sum\nolimits_{i,j} c_{ij} x_{ij} + \sum\nolimits_j \lambda_j (1 - \sum\nolimits_i x_{ij})$$
$$\text{s.t.} \quad \sum\nolimits_j a_{ij} x_{ij} \leq b_i, \ \forall i \quad (\mathrm{KP})$$
$$= \sum\nolimits_j \lambda_j + \sum\nolimits_i \{\mathrm{Min} \sum\nolimits_j (c_{ij} - \lambda_j) x_{ij} \Big| \sum\nolimits_j a_{ij} x_{ij} \leq b_i, \forall i$$
$$x_{ij} \in \{0, 1\}, \ \forall j\},$$

and (LR) is the maximum with respect to λ of $\mathrm{v}(\mathrm{LR}_\lambda)$.

Let $\mathrm{EP(KP)} = \{x^k | k \in K\}$ be the set of all integer feasible solutions of the constraints (KP), and let $\mathrm{EP(KP_i)} = \{x_i^k | k \in K_i\}$ be the set of all integer feasible solution of the i^{th} knapsack, with $K = \prod_i K_i$. Then a feasible solution of (LR_λ) can be described by $x_{ij} = \sum_{k \in K_i} \mu_k^i x_{ij}^k, \forall i, j$.

The Lagrangian dual is equivalent to the aggregate master problem AMP:

$$\begin{aligned}&(\mathrm{AMP})\ \mathrm{Max}_{\lambda,\zeta}\left\{\zeta | \zeta \le \sum_{i,j} c_{ij} x_{ij}^k + \sum_j \lambda_j (1 - \sum_i x_{ij}^k), \forall k \in K\right\}\\ &= \mathrm{Max}_{\lambda,z}\left\{z + \sum_j \lambda_j | z \le \sum_{ij} (c_{ij} - \lambda_j) x_{ij}^k,\ \forall k \in K\right\}\end{aligned}$$

with the substitution $\zeta = z + \sum_j \lambda_j$.

If one had first written the column generation formulation for the Lagrangian dual, one would naturally have de-coupled the solutions of the independent knapsack subproblems, using the independent sets K_i instead of K, the column generation master problem would have been disaggegated:

$$\begin{aligned}&(\mathrm{DMP})\mathrm{Max}_{\lambda,z} \sum_i z_i + \sum_j \lambda_j\\ &\qquad \text{s.t.} \quad z_i \le \sum_j (c_{ij} - \lambda_j) x_{ij}^k,\ \forall i, \forall k \in K_i\end{aligned}$$

and its dual

$$\begin{aligned}\mathrm{Min}_\mu \sum_{k \in K_i} \sum_{i,j} c_{ij} x_{ij}^k \mu_k^{(i)} \Big| &\sum_{k \in K_i} \sum_i x_{ij}^k \mu_k^{(i)} = 1, \forall j,\\ &\mu_k^i \ge 0, \sum_{k \in K_i} \mu_k^{(i)} = 1, \forall i\},\end{aligned}$$

is clearly the Dantzig-Wolfe decomposition of the primal equivalent

$$(\mathrm{PR}) \quad \mathrm{Min}_x \left\{\sum_{i,j} c_{ij} x_{ij} \Big| \sum_i x_{i,j} = 1,\ x_{ij} \ge 0\right\}$$

of (LR).

Relax-and-Cut

One question that often arises in the context of Lagrangian relaxation is how to strengthen the Lagrangian relaxation bound. One possible answer is the addition of cuts that are currently violated by the Lagrangian solution. It is clear however that adding these to the Lagrangian problem will change its structure and may make it much harder to solve. One possible way out is to dualize these cuts (for a more detailed analysis, see (Guignard 1998)). Remember that dualizing does not mean discarding! The cuts will be added to the set of complicating constraints, and intuitively they will be useful only if the intersection NI (for "new intersection") of the new relaxed polyhedron and of the convex hull of the integer solutions of the kept constraints is "smaller" than the intersection OI (for "old intersection") of the old relaxed polyhedron and of the convex hull of the integer solutions of the kept constraints. This in turn is only possible if the new relaxed polyhedron is smaller than the old one, since the kept constraints are the same in both cases. This has the following implications. Consider a cut that is violated by the current Lagrangian solution:

(1) if the cut is just a convex combination of the current constraints, dualized and/or kept, it cannot possibly reduce the intersection, since every point of the "old" intersection will also satisfy it; so in particular surrogate constraints of the dualized constraints cannot help.
(2) if the cut is a valid inequality for the Lagrangian problem, then every point in the convex hull of the integer points of the kept constraints satisfies it, because every integer feasible solution of the Lagrangian subproblem does;
(3) it is thus necessary for the cut to use "integer" information from both the dualized and the kept constraints, and to remove part of the intersection. (Remember that the Lagrangian solution is an integer point required to satisfy only the kept constraints).

A Relax-and-Cut scheme could proceed as follows:

1. Initialize the Lagrangian multiplier λ.
2. Solve the current Lagrangian problem, let $x(\lambda)$ be the Lagrangian solution. If the Lagrangian dual is not solved yet, update λ. Else end.
3. Identify a cut that is violated by $x(\lambda)$, and dualize it. Go back to 2.

The term Relax-and-Cut was first used by (Escudero et al. 1994). In that paper, a partial description of the constraint set was used, and violated constraints (not cuts) were identified, added to the model and immediately dualized. The idea, if not the name, had actually been used earlier. For instance

in solving TSP problems, subtour elimination constraints were generated on the fly and immediately dualized in Balas and Christofides (1981). The usefulness of constraints is obvious, contrary to that of cuts. A missing constraint can obviously change the problem solution.

Here are examples of cuts that if dualized cannot possibly tighten Lagrangian relaxation bounds.

Non-improving Dualized Cuts: Example for the GAP

Consider again the GAP model.

If one dualizes (MC), the Lagrangian relaxation problem decomposes into one subproblem per *j:*

$$
\begin{aligned}
(\mathrm{LR}_\lambda)\ \mathrm{Min} & \sum_{i,j} c_{ij}x_{ij} + \sum_j \lambda_j\Big(1 - \sum_i x_{ij}\Big) \\
\text{s.t.} \quad & \sum_j a_{ij}x_{ij} \le b_i, \quad \forall i \quad (\mathrm{KP}) \\
= \mathrm{Min}\ & \Big\{\sum_{i,j}(c_{ij} - \lambda_j)x_{ij} + \sum_j \lambda_j \Big| \\
& \sum_j a_{ij}x_{ij} \le b_i, \quad \forall i, \quad x_{ij} \in \{0,\ 1\}, \forall i, j\Big\} \\
= \sum_j \lambda_j + \sum_i & \Big\{\mathrm{Min} \sum_j (c_{ij} - \lambda_j)\ x_{ij} \Big| \\
& \sum_j a_{ij}x_{ij} \le b_i, \quad \forall i\, x_{ij} \in \{0, 1\}, \quad \forall j\Big\}.
\end{aligned}
$$

Thus the i^{th} Lagrangian subproblem is a knapsack problem for the i^{th} machine. After solving all knapsack problems, the solution $x(\lambda)$ may violate some multiple choice constraint, i.e., there may exist some j for which $\sum_i x_{ij} \neq 1$, and as a consequence the condition $\sum_i \sum_j x_{ij} = |J|$ may be violated. Adding this "cut" (it indeed cuts out the current Lagrangian solution!), and immediately dualizing it, does not reduce the intersection, as every point of the old intersection OI already satisfies all multiple choice constraints (MC), i.e., the dualized constraints.

Can kept Cuts Strengthen the Lagrangian Bound?

What happens if one keeps the cuts instead of dualizing them? It is clear that adding these to the Lagrangian problem will change its structure, but it may still be solvable rather easily. The cuts will be added to the set of easy constraints, and intuitively they will be useful only if the intersection NI (for "new intersection") of the relaxed polyhedron and of the new convex hull of the integer solutions of the kept constraints is smaller than the intersection OI (for "old intersection") of the relaxed polyhedron and of the old convex hull of the integer solutions of the kept constraints. This in turn is only possible if the new convex hull polyhedron is smaller than the old one, since the dualized constraints are the same in both cases.

Example 6. Consider again the GAP, and its weak Lagrangian relaxation in which the knapsack constraints (KP) are dualized. One could add to the remaining multiple choice constraints a surrogate constraint of the dualized constraints, for instance the sum of all knapsack constraints, which is obviously weaker than the original knapsack constraints. The Lagrangian problem does not decompose anymore, but its new structure is that of a multiple choice knapsack problem, which is usually easy to solve with specialized software, and much easier than the aggregate knapsack without multiple choice constraints. The above strengthening of the Lagrangian bound is simple, yet potentially powerful.

Lagrangian Heuristics and Branch-and-Price

Lagrangian relaxation provides bounds, but it also generates Lagrangian solutions. If a Lagrangian solution is feasible and satisfies complementary slackness (CS), one knows that it is an optimal solution of the IP problem. If it is feasible but CS does not hold, it is at least a feasible solution of the IP problem and one still has to determine, by BB or otherwise, whether it is optimal. Otherwise, Lagrangian relaxation generates infeasible integer **solutions**. Yet quite often these solutions are nearly feasible, as one got penalized for large constraints violations. There exists a very large body of literature dealing with possible ways of modifying existing infeasible Lagrangian solutions to make them feasible. Lagrangian heuristics are essentially problem dependent. Here are a few hints on how one may want to proceed. One may for instance try to get **feasible** solutions in the following way:

(1) by modifying the solution to correct its infeasibilities while keeping the objective function deterioration small.

Example: in production scheduling, if one relaxes the demand constraints, one may try to change production levels (down or up) so as to meet the demand (de Matta and Guignard 1994).

(2) by fixing (at 1 or 0) some of the meaningful decision variables according to their value in the current Lagrangian solution, and solving optimally

the remaining problem. Chajakis et al. (1996) called this generic approach the lazy Lagrangian heuristic. One guiding principle may be to fix variables that satisfy relaxed constraints.

Part of the success of Lagrangian relaxation comes from clever implementations of methods for solving the Lagrangian dual, with powerful heuristic imbedded at every iteration. In many cases, the remaining duality gap, i.e., the relative percentage gap between the best Lagrangian bound found and the best feasible solution found by heuristics is sufficiently small to forego enumeration. In some instances however an optimal or almost optimal solution is desired, and a Branch-and-Bound scheme adapted to replace LP bounds by LR bounds can be used. If the Lagrangian dual is solved by column generation, the scheme is called Branch-and-Price, as new columns may need to be priced-out as one keeps branching see Desrosiers et al. 1984), (Barnhart et al., 1998). In that case, branching rules need to be carefully designed. The hope is that such schemes will converge faster than LP-based Branch-and-Bound, as bounds will normally be tighter and nodes may be pruned faster. The amount of work done at a node, though, may be substantially more than solving an LP.

Concluding Remarks

- Lagrangian relaxation is a powerful family of tools for solving approximately integer programming problems. It provides
  - stronger bounds than LP relaxation when the problem(s) don't have the Integrality Property.
  - good starting points for heuristic search.
- The availability of powerful interfaces (GAMS, AMPL, etc.) and of flexible IP packages makes it possible for the user to try various schemes and to implement and test them.
- As illustrated by the varied examples described in this paper, Lagrangian relaxation is very flexible. Often some reformulation is necessary for a really good scheme to appear.
- It is not necessary to have special structures embedded in a problem to try to use Lagrangian schemes. If it is possible to decompose the problem structurally into meaningful components and to split them through constraint dualization, possibly after having introduced new variable expressions, it is probably worth trying.
- Finally, solutions to one or more of the Lagrangian subproblems might lend themselves to Lagrangian heuristics, possibly followed by interchange heuristics, to obtain good feasible solutions.

Lagrangian relaxation bounds coupled with Lagrangian heuristics provide the analyst with brackets around the optimal integer value. These are usually much tighter than the brackets coming from LP-based bounds and heuristics

See

- ▶ Branch and Bound
- ▶ Convex Hull
- ▶ Convex Optimization
- ▶ Heuristics
- ▶ Integer and Combinatorial Optimization
- ▶ Traveling Salesman Problem

References

Andalaft, N., Andalaft, P., Guignard, M., Magendzo, A., Wainer, A., & Weintraub, A. (2003). A problem of forest harvesting and road building solved through model strengthening and Lagrangean relaxation. *Operations Research, 51*(4), 613–628.

Balas, E., & Christofides, N. (1981). A restricted Lagrangean approach to the traveling salesman problem. *Mathematical Programming, 21*, 19–46.

Barahona, F., & Anbil, R. (2000). The volume algorithm: Producing primal solutions with a subgradient method. *Mathematical Programming, 87*(3), 385–399.

Barnhart, C., Johnson, E. L., Nemhauser, G. L., Savelsbergh, M. W. P., & Vance, P. (1998). Branch-and-price: Column generation for solving huge integer programs. *Operations Research, 46*(3), 316–329.

Bilde, O., & Krarup, J. (1977). Sharp lower bounds and efficient algorithms for the simple plant location problem. *Annals of Discrete Mathematics, 1*, 79–97. Also a 1967 report in Danish Bestemmelse af optimal beliggenhed af produktionssteder, Research Report, IMSOR.

Birge, J., & Louveaux, F. (2011). *Introduction to stochastic programming* (Springer series in operations research and financial engineering, 2nd ed.). New York: Springer.

Chajakis, E., Guignard, M., Yan, M., & Zhu, S. (1996). The Lazy Lagrangean heuristic. Optimization Days, Montreal, May 1996.

Chen, B., & Guignard, M. (1998). Polyhedral analysis and decompositions for capacitated plant location-type problems. *Discrete Applied Mathematics, 82*, 79–91.

Dantzig, G. B., & Wolfe, P. (1960). The decomposition principle for linear programs. *Operations Research, 8*, 101–111.

Dantzig, G. B., & Wolfe, P. (1961). The decomposition algorithm for linear programs. *Econometrica, 29*(4), 767–778.

de Matta, R., & Guignard, M. (1994). Dynamic production scheduling for a process industry. *Operations Research, 42*, 492–503.
Desrosiers, J., Soumis, F., & Desrochers, M. (1984). Routing with time windows by column generation. *Networks, 14*(4), 545–565.
Escudero, L. F. (2009). On a mixture of the fix-and-relax coordination and Lagrangean substitution schemes for multistage stochastic mixed integer programming. *TOP, 17*, 5–29.
Escudero, L., Guignard, M., & Malik, K. (1994). A Lagrangean relax-and-cut approach for the sequential ordering problem with precedence relationships. In C. Ribeiro (Ed.), *Annals of operations research*, 50, Applications of combinatorial optimization, pp. 219–237.
Everett, H., III. (1963). Generalized Lagrange multiplier method for solving problems of optimum allocation of resources. *Operations Research, 11*, 399–417.
Fisher, M. L. (1981). The Lagrangian relaxation method for solving integer programming problems. *Management Science, 27*, 1–18.
Fisher, M. L. (1985). An applications oriented guide to Lagrangian relaxation. *Interfaces, 15*, 10–21.
Geoffrion, A. M. (1974). Lagrangean relaxation for integer programming. *Mathematical Programming Study, 2*, 82–114.
Geoffrion, A. M., & McBride, R. (1978). Lagrangean relaxation applied to capacitated facility location problems. *AIIE Transactions, 10*, 40–47.
Glover, F., & Klingman, D. (1988). Layering strategies for creating exploitable structure in linear and integer programs. *Mathematical Programming, 40*(2), 165–182.
Guignard, M. (1998). Efficient cuts in Lagrangean relax-and-cut schemes. *European Journal of Operational Research, 105*(1), 216–223.
Guignard, M., & Kim, S. (1987). Lagrangean decomposition: a model yielding stronger Lagrangean bounds. *Mathematical Programming, 39*, 215–228.
Guignard, M., & Rosenwein, M. B. (1989). An application-oriented guide for designing Lagrangian dual ascent algorithms. *European Journal of Operational Research, 43*, 197–205.
Guignard, M., & Yan, H. (1993). Structural decomposition methods for dynamic multi-hydropower plant optimization. *Research report 93-12-01*, Operations and Information Management Department, University of Pennsylvania.
Held, M., & Karp, R. M. (1970). The traveling salesman problem and minimum spanning trees. *Operations Research, 18*, 1138–1162.
Held, M., & Karp, R. M. (1971). The traveling salesman problem and minimum spanning trees: Part II. *Mathematical Programming, 1*, 6–25.
Lemaréchal, C. (1974). An algorithm for minimizing convex functions. In *Proceedings IFIP'74 congress*, North Holland, Amsterdam, pp. 552–556.
Näsberg, M., & Jörnsten, K. O., & Smeds, P. A. (1985). Variable splitting – A new Lagrangean relaxation approach to some mathematical programming problems. *Report LITH-MAT-R-85-04*, Linkoping University.
Reinoso, H., & Maculan, N. (1992). Lagrangean decomposition in integer linear programming: A new scheme. *INFOR 30*(1).
Soenen, R. (1977). *Contribution à l'étude des systèmes de conduite en temps réel en vue de la commande d'unités de fabrication*. Thèse de Doctorat d'Etat, Université de Lille, France.

Lanchester Attrition

The concept of an explicit mathematical relationship between opposing military forces and casualty rates. The two classical laws are the linear law, that gives the casualty rate (derivative of force size with respect to time) of one side as a negative constant multiplied by the product of the two sides' force sizes, and the square law, which gives the casualty rate of one side as a negative constant multiplied by the opposing side's force size.

See

▶ Battle Modeling
▶ Homogeneous Lanchester Equations
▶ Lanchester's Equations

Lanchester's Equations

Joseph H. Engel
Bethesda, MD, USA

Introduction

Lanchester's equations are named for the Englishman, F.W. Lanchester, who formulated and presented them in 1914 in a series of articles contributed to the British journal, *Engineering*, which then were printed in toto in Lanchester (1916). More recent presentation of these results appeared in the 1946 Operations Evaluation Group Report No. 54, *Methods of Operations Research* by Philip M. Morse and George E. Kimball, which was published commercially by John Wiley and Sons (Morse and Kimball 1951). In addition, a reprint of the original 1916 Lanchester work, "Mathematics in Warfare," appeared in *The World of Mathematics*, Vol. 4, prepared by James R. Newman and published by Simon and Schuster in 1956.

The significance of these equations is that they represented possibly the first mathematical analysis of forces in combat, and served as the guiding light (for the U.S. and its allies) behind the development,

during and after World War II, of all two sided combat models, simulations, and other methods of calculating combat losses during a battle.

It appears that M. Osipov developed and published comparable equations in a Tsarist Russian military journal in 1915, perhaps independent of Lanchester's results. A translation of his work into English, prepared by Robert L. Helmbold and Allen S. Rehm, was printed in September 1991 by the U.S. Army Concepts Analysis Agency.

Lanchester's equations present a mathematical discussion of concepts such as the relative strengths of opposing forces in battle, the nature of the weapons, the importance of concentration, and their effects on casualties, and the outcome of the battle. His arguments are paraphrased here, preserving much of his original symbolism. The equations deal with ancient warfare and modern warfare.

Ancient Warfare

Lanchester explained that, because of the limited range of weapons in ancient warfare (like swords), the number of troops on one side of a battle (the Blue force) that are actively engaged in hand-to-hand combat on the combat front at any time during the battle must equal approximately the number of troops responding to them on the other side (the Red force). For this reason, one may assume that the rate at which casualties are produced is constant, because the number of troops actively engaged on each side is constant (until very near the end of the battle), and the rate c (>0), at which Blue combatants become casualties is a product of the fixed number of Red troops engaged and their average individual casualty producing effectiveness (dependent on the average strength of Red's weapons and the effectiveness of the Blue defenses). Similar results apply to k (>0), the Red casualty rate. The two casualty rates need not be the same, as the weapons and defenses of the two sides may differ.

If $b(t)$ is the number of effective Blue troops at time t after the battle has started and $r(t)$ is the number of effective Red troops, the following equations can be assumed to obtain:

$$db/dt = -c, dr/dt = -k. \tag{1}$$

The relationship between the sizes of the two forces may easily be ascertained by observing from (1) that

$$db/dr = c/k, \tag{2}$$

from which it can be deduced that

$$k[b(0) - b(t)] = c[r(0) - r(t)]. \tag{3}$$

In the above equations, $b(0)$ and $r(0)$ are assumed to be the initial (positive) sizes of the forces at time 0, the beginning of the battle, and the equations are valid only as long as $b(t)$ and $r(t)$ remain greater than zero. Assuming the combatants battle until all the troops on one side or the other are useless for combat, having become casualties, the battle ends at the earliest time when $b(t)$ or $r(t)$ becomes equal to zero. Thus, solving for r in (3) when b becomes 0 (or vice versa) yields: when

$$b(t) = 0, r(t) = [c^* r(0) - k^* b(0)]/c$$

and when

$$r(t) = 0, b(t) = [k^* b(0) - c^* r(0)]/k. \tag{4}$$

Thus, if $c^* r(0) > k^* b(0)$, the Red force wins the battle, while if $k^*\ b(0) > c^*\ b(0)$, the Blue force wins the battle. Summarizing these observations by designating the initial effectiveness of the Blue force to be $k^* b(0)$, and that of the Red force $c^* r(0)$, shows that the force with the larger initial effectiveness wins, while equal initial effectiveness ensures a draw.

It is also simple to return to the original differential equations of (1) and to solve them to determine the number of effective troops of either force as a linear function of time. This essentially completes Lanchester's modeling of ancient warfare.

Modern Warfare

Lanchester postulated that the major difference between modern and ancient warfare is the ability of modern weapons (such as rifles and, to a lesser degree bows and arrows, cross bows, etc.) to produce casualties at long range. As a result, the troops on one side of an engagement can, in principle, be fired upon by the entire opposing force. Consequently, assuming

that all of each of the troops on a side have the same (average) ability to produce casualties at a fixed rate, the combined casualty rate against a given side is proportional to the number of effective troops on the other side.

This leads directly to the following differential equations constituting Lanchester's model of modern warfare:

$$db/dt = -c^*r, dr/dt = -k^*b. \quad (5)$$

As in the ancient warfare case, the individual casualty producing rates, c and k, are assumed to be known constants for the duration of the battle.

Now combine these two equations (as was done in the ancient warfare case) and obtain

$$db/dr = (c^*r)/k^*b. \quad (6)$$

Equation (6) is solved to obtain the relationship between the numbers of effective forces on the two sides as the battle progresses. This leads to

$$k\left[b^2(0) - b^2\right] = c\left[r^2(0) - r^2\right]. \quad (7)$$

Since these equations are valid only when $b \geq 0$ and $r \geq 0$, observe, as in the ancient warfare case, that, with the battle ending when the losing side has been reduced through casualties to no effective troops, and the victor has a positive number of effective troops, the force with the larger initial effectiveness, [$k^*b^2(0)$ for Blue and $c^*r^2(0)$ for Red], will win the battle, while equal initial effectiveness produces a draw. Equation (7) and this paragraph constitute Lanchester's Square Law for his model of modern warfare.

Again, as in the ancient warfare case, it is possible to solve the initial differential equations in (5) to obtain the specific functions that describe the behavior of the side of either force as a function of time. These results also appear in Morse and Kimball, (1951), and this essentially completes Lanchester's modeling of modern warfare.

Extensions

In presenting his results, Lanchester used many techniques that are taken for granted in contemporary OR practice. He formulated clear assumptions about the operation of the system he was studying, derived the mathematical consequences of his assumptions, and discussed how variation of assumptions affected results. Consequently he was able to provide specific numerical insights into characteristics of the system that could be translated into useful ways of improving a system that operated in accordance with the specified assumptions.

It was possible for Lanchester to accomplish his mathematical modeling by using what is often referred to as the First Theorem of Operations Research:

> A function of the average equals the average of the function.

The above result applies only in very special circumstances; nevertheless, there are many cases in which use of this theorem allows deterministic results to be derived easily. Such results will usually provide a good approximation of average results occurring in reality. It is through this technique that various chemical formulas or formulas in the physical sciences pertaining to concepts such as temperature, thermodynamics, etc., were derived.

In those formulations, it is assumed that a group of many small objects moving at various speeds with a known average speed will function in the same manner as if all the objects moved at the same (average) speed. Similarly, in his warfare modeling, Lanchester assumed that the casualty producing rate of every one of the troops on one side of a battle was constant and equal to the average (per troop) casualty producing rate of the entire force, and the same is true of the troops on the other side.

The usefulness of Lanchester's work is primarily in its demonstration of the fact that it is possible to draw mathematical and numerical conclusions concerning the occurrence of casualties in certain battles that can be described, a priori, as conforming to certain specified assumptions concerning how the battle is conducted. From such an observation, it is possible to generalize and derive other models that conform to other sets of assumptions, so that a wider range of combat situations can be dealt with. This has led to all sorts of models that can be handled through generalizations of Lanchester's techniques.

The analyst can take into account other factors not specifically covered by Lanchester, such as addition or

withdrawal of troops in the course of an engagement. Movement of forces can be considered. Different weapons and defensive techniques can be studied.

Dispersing and hiding the troops on one side of a battle (as in guerrilla warfare) affects the rate at which they can be hit by the other side, which led Lanchester to present another differential equation for such a force. This leads to analyses in which one or the other or both forces engage in ancient, modern, or guerrilla warfare. There are nine kinds of battles that an analyst can deal with just by adding the consideration of the possibility of guerrilla warfare to his bag of tricks (Deitchman 1962).

Clearly there is a great deal of flexibility in deriving models involving the use of deterministic differential equations that predict specific average results. The probabilistic events that take place during the course of a battle can also be dealt with in comparatively simple cases as demonstrated by B.O. Koopman and described in Morse and Kimball (1951). Regrettably, the mathematics of probabilistic systems is frequently much more difficult than that of deterministic systems, and the need to recognize the existence of all sorts of complications in a battle, frequently leads to rather complicated and abstruse mathematics which can best be handled through the use of computers for the required numerical calculations.

The field of combat simulation is recognized as a direct descendant of the Lanchester approach. Of historic interest in this connection is the fact that Lt. Fiske of the U.S. Navy presented, in 1911, a model of warfare consisting of a salvo by salvo table that computed casualties on two sides of a battle. This material was brought to the attention of contemporary analysts by H.K. Weiss (1962).

Engel (1963) showed that the equations of the Fiske model were difference equations that became, in the limit as the time increment between successive salvos approached zero, identical to the Lanchester differential equations of modern warfare. In a sense, this validated the use of discrete time models that approximated combat models for computer calculations, allowing greater confidence on the part of the analyst that no great surprises would result from a use of such discrete time approximations of combat models.

A cautionary note must be sounded at this point. Before using whatever mathematical model the analyst may have derived in discussing any past or future battles, the analyst must be certain that the assumptions of the model on how the battle will be conducted and terminated pertain to the battle being analyzed. The analyst should be able to derive the appropriate values of any parameters (such as $b(0)$, $r(0)$, c and k) to be used in the Lanchester or other models believed to apply in the case under study. Thought experiments do not suffice. The analyst must examine data to determine whether the assumptions provide a valid description of the way the battle proceeds, and to ascertain from relevant combat and experimental data that the model's numerical values for the parameters are appropriate.

Validation of Equations

Lanchester did not provide any demonstration of the relevance of his models to any specific historic battles, although he did discuss examples from history in which he suggested that the results of certain tactical actions were consistent with results that could be derived from his models. A validation of Lanchester's modern warfare equations was first given by Engel (1954), based on an analysis of the Battle of Iwo Jima during World War II. The analysis showed that the daily casualties inflicted on the U.S. forces over the approximately forty days of the battle were consistent with Lanchester's model for modern warfare. Since that time, additional analyses of combat results and experiments have demonstrated that the values of various parameters can be estimated for use in specified combat situations, and that appropriate combat models can be used in conjunction with those parameter values to obtain results of interest to military planners and decision makers.

The modeling methodology pioneered by Lanchester in the field of combat casualty analysis has served as a most important guide for analysts of military problems. He showed how application of these techniques can be used in developing mathematical models of combat that can be applied in forecasting the results of hypothetical battles. This enables operations research analysts to predict outcomes of these battles, plan tactics and strategy, develop weapons requirements, determine force requirements, and otherwise assist planners and decision makers concerned with the effective use of military forces.

See

- ▶ Battle Modeling
- ▶ Military Operations Research
- ▶ Verification, Validation, and Testing of Models

References

Deitchman, S. J. (1962). A Lanchester model of guerrilla warfare. *Operations Research, 10*, 818–827.
Engel, J. H. (1954). A verification of Lanchester's Law. *Journal of the Operations Research Society of America, 2*, 163–171.
Engel, J. H. (1963). Comments on a paper by H.K. Weiss. *Operations Research, 11*, 147–150.
Lanchester, F. W. (1916). *Aircraft in warfare: The dawn of the fourth arm*. London: Constable and Company.
Morse, P. M., & Kimball, G. E. (1951). *Methods of operations research*. New York: Wiley. Also Dover Publications, 2003.
Weiss, H. K. (1962). The Fiske model of warfare. *Operations Research, 10*, 569–571.

Laplace Transform

For any function $g(t)$ defined on $t \geq 0$ (e.g., a probability density), its Laplace transform is defined as $\int_0^\infty e^{-st} g(t)dt$, $\mathrm{Re}(s) > 0$.

Laplace-Stieltjes Transform

For any function $G(t)$ defined on $t \geq 0$ (e.g., a cumulative probability distribution function), its Laplace-Stieltjes transform (LST) is defined as $\int_0^\infty e^{-st} dG(t), \mathrm{Re}(s) > 0$. When the function $G(t)$ is differentiable, it follows that the LST is equivalent to the regular Laplace transform of the derivative, say $g(t) = dG(t)/dt$.

Large Deviations

In probability theory, the study of asymptotic tail behavior of sequences of probability distributions. For example, the probability that a sample mean exceeds a certain threshold decays exponentially to zero according to some rate function. Large deviations theory is used in stochastic simulation for more effectively estimating rarely occurring events.

See

- ▶ Rare Event Simulation

References

Dembo, A., & Zeutoni, O. (2009). *Large deviations techniques and applications* (2nd ed.). New York: Springer.
Varadhan, S. R. S. (2008). Special invited paper: Large deviations. *Annals of Probability, 36*(2), 397–419.

Large-Scale Systems

James K. Ho
University of Illinois at Chicago, Chicago, IL, USA

Introduction

In OR/MS, large-scale systems refer to the methodology for the modeling and optimization of problems that, due to their size and information content, challenge the capability of existing solution technology (Lasdon 1970). There is no absolute measure to classify such problems. In any given computing environment, the cost-effectiveness of problem solving generally depends on the dimensions and the volume of data involved. As problems get larger, the cost tends to go up, lowering effectiveness. Even before the physical limits of the hardware or the numerical resolution of the software are exceeded, the effectiveness of the solution environment may have become unacceptable. Efforts to improve on any of the relative performance measures such as solution time, numerical accuracy, memory and other resource requirements, are subjects in the topic of large-scale systems. Since solving larger problems more effectively is also an obvious goal in all specializations of operations research, there are natural linkages and necessary overlaps with most other areas in the field (Nemhauser 1994).

All known methodology for large-scale systems can be viewed as the design of computational techniques to take advantage of various structural properties exhibited by both the problems and known solution algorithms (Koussoulas and Groumpos 1999). Broadly speaking, such special properties can be regarded as either micro-structures or macro-structures. Micro-structures are properties that are independent of permutations in the ordering of the variables and constraints in the problem. An example is sparsity in the constraint coefficients. Macro-structures are those that depend on such orderings. An example is the block structure of loosely coupled or dynamic systems.

Using Micro-Structures of Problems

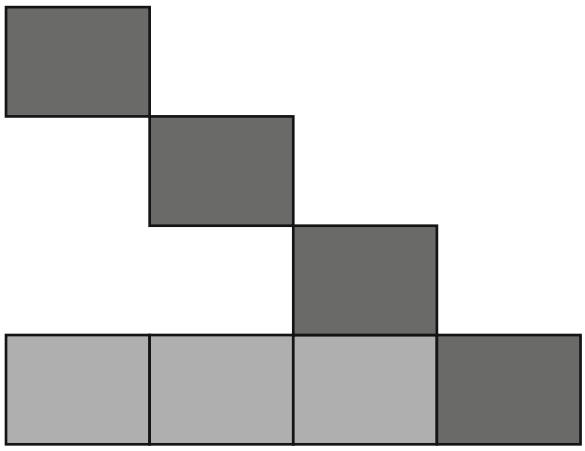
In the modeling of real systems, the larger the problem, the less likely it is for a variable to interact with all the others. If each variable is coupled only to a small subset of the total, the resulting constraints will be sparse. Techniques that eliminate the representation of the nonexistent interactions can reduce storage requirement significantly. For example, a linear program with 10,000 variables and 10,000 constraints has potentially 10^8 coefficients. If on the average, each variable appears in 10 constraints, there will be only 10^5 nonzero coefficients, implying a density of 0.1%. Sparse matrix methods from numerical analysis have been used with great success here. Furthermore, the nonzero coefficients may come from an even smaller pool of unique values. This feature is known as supersparsity and allows additional economy in data storage. Large, complex models are usually generated systematically by applying the logic of the problem iteratively over myriad parameter sets. This may lead to formulations with redundant variables and constraints. Examples include flow balance equations that produce a redundant constraint when total input equals total output; lower and upper bounds that are equal imply the variable can be fixed. Methods to simplify the problem by identifying and removing such redundancies are incorporated into the procedure of preprocessing. It is not unusual to observe reductions of problem dimensions by 10 to 50% with this approach.

Using Micro-Structures of Algorithms

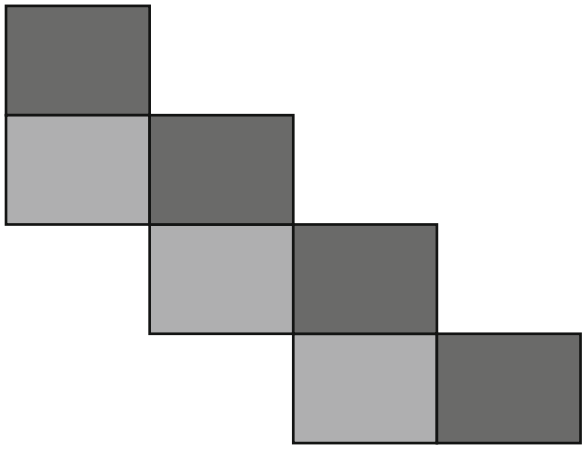
Algorithms may have steps that are adaptable to advanced computing architecture at the micro-processing level. An example is the vectorization of inner-product calculations in the simplex method. A completely different exploit is the relatively low number of iterations required by interior-point methods. As the number of iterations seems to grow rather slowly with problem size, it is a micro-structure of such algorithms that automatically sheds light on the optimization of large-scale systems. Yet another promising approach that falls under this heading is the use of sampling techniques in stochastic optimization.

Using Macro-Structures of Problems

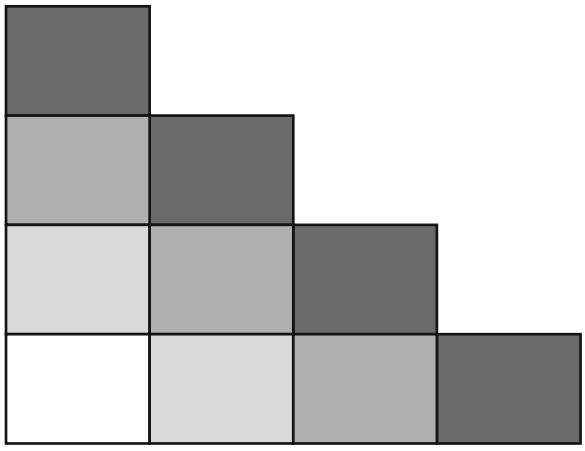
Most large-scale systems are comprised of interacting subsystems. Examples are multidivisional firms with a headquarters coordinating the activities of the semi-autonomous divisions; time-phased models of dynamic systems with linkages only among adjacent time periods; capital investment or financial planning models with each period linked to all subsequent periods. Linear programming modeling of the above examples gives rise to problems with the block-angular, staircase and block-triangular structures, respectively (Figs. 1, 2, and 3). Other variations and combinations are also possible. Two major approaches to take advantage of such structures are decomposition and factorization. Decomposition relies on algorithms that transform the problem into a sequence of smaller subproblems that can be solved independently. Various schemes are devised to coordinate the subproblems and steer them towards the overall solution. Many algorithms are derived from the Dantzig-Wolfe decomposition principle which provides a rigorous framework for this approach. Factorization is the adaptation of existing algorithms to take advantage of the problem structure. In the case of the simplex method, the representation of the basis matrix required at each step can be partitioned into blocks and updated separately. It has been shown that all of the simplex-based techniques proposed over the years under somewhat confusing guises



Large-Scale Systems, Fig. 1 Block-angular structure



Large-Scale Systems, Fig. 2 Staircase structure



Large-Scale Systems, Fig. 3 Block-triangular structure

of partitioning and decomposition are indeed special cases of the factorization approach (Dantzig et al. 1981).

Using Macro-Structures of Algorithms

Both decomposition and factorization algorithms are natural candidates for parallel and distributed computation since they involve the solution of independent subproblems. The latter can be solved concurrently on multiprocessor computers of various architectures. Particularly suitable is the class of Multiple-Instruction-Multiple-Data (MIND) machines that are essentially networks of processors that can execute independent instructions. They represent a cost-effective way to harness tremendous computing power from relatively modest and economical components. One processor can be programmed as the coordinator of the algorithmic procedures. Each of the other processors can be assigned a subproblem and programmed to communicate with the coordinating process. As the gain in overall efficiency is bounded by the number of processors used, the intent of this approach is to realize the full potential of certain algorithms rather than fundamentally enhancing their performance. It is, however, becoming an essential aspect of large-scale systems, as multi-processor computers are expected to be prevalent (Eckstein 1993). Early results have been obtained for decomposition (Ho and Sundarraj 1997), factorization (Ho and Sundarraj 1994), and barrier methods (Lustig and Rothberg 1996).

Concluding Remarks

Linear and mixed integer programming remain the primary focus in the optimization of large-scale systems. New computer architectures with ever-increasing processing power and memory capacities have facilitated the empirical approach to algorithmic development. Experimentation with large-scale problems becomes a viable strategy to identify, test, and fine tune ideas for improvement. This has been especially successful in commercial implementations of both the simplex and interior-point methods exploiting mainly the micro-structures of problems and algorithms. Problems with hundreds of

thousands of constraints and millions of variables are solvable on workstation-grade computers (Fourer 2009). Earlier experiences with macro-techniques in decomposition and factorization did not have the benefits of the more modern technological advances. The results are either inconclusive or less than promising (Ho 1987). Future work, especially in hybrid schemes using advanced hardware, may lead to significant contributions to large-scale non-linear, integer and stochastic optimization.

See

- ▶ Dantzig-Wolfe Decomposition Algorithm
- ▶ Density
- ▶ Integer and Combinatorial Optimization
- ▶ Linear Programming
- ▶ Nonlinear Programming
- ▶ Parallel Computing
- ▶ Sparsity
- ▶ Super-Sparsity

References

Dantzig, G. B., Dempster, M. A. H., & Kallio, M. J., (Eds.) (1981). Large-scale linear programming. IIASA CP-81-S1, Laxenburg, Austria.

Eckstein, J. (1993). Large-scale parallel computing, optimization and operations research: A survey. ORSA Computer Science Technical Section Newsletter, 14, Fall.

Fourer, R. (2009). 2009 Linear programming software survey. OR/MS Today, 5 June 2009.

Ho, J. K. (1987). Recent advances in the decomposition approach to linear programming. *Mathematical Programming Study, 31*, 119–128.

Ho, J. K., & Sundarraj, R. P. (1994). On the efficacy of distributed simplex algorithms for linear programming. *Computational Optimization and Applications, 3*, 349–363.

Ho, J. K., & Sundarraj, R. P. (1997). Distributed nested decomposition of staircase linear programs. *ACM Transactions on Mathematical Software, 23*, 148–173.

Koussoulas, N. T., & Groumpos, P. P., (Eds.) (1999). Large scale systems: Theory and applications. In *Proceedings of the 8th IFAC/IFORS/IMACS/IFIP symposium*, July 1998, Elsevier Science, Amsterdam.

Lasdon, L. S. (1970). *Optimization theory for large systems*. New York: MacMillan.

Lustig, I. J., & Rothberg, E. (1996). Gigaflops in linear programming. *Operations Research Letters, 18*, 157–165.

Nemhauser, G. L. (1994). The age of optimization: Solving large-scale real-world problems. *Operations Research, 42*, 5–13.

Las Vegas Algorithm

Randomized algorithm that is guaranteed to give the correct result 100% of the time, in contrast to Monte Carlo methods, which provide statistical bounds.

See

- ▶ Monte Carlo Methods
- ▶ Randomized Algorithm

References

Hromkovic, J. (2005). *Design and analysis of randomized algorithms*. New York: Springer.

Latest Finish Time

The latest time an activity must be completed without delaying the end of a project. It is simply the sum of the latest start time of the activity and its duration.

See

- ▶ Network Planning

Latest Start Time

The latest time an activity can start without delaying the end of a project. A delay of an activity beyond the latest start time will delay the entire project completion by a corresponding amount. These times are calculated on the basis of a reverse pass through the network.

See

- ▶ Network Planning

Latin Square

▶ Combinatorics

LCFS

A queueing discipline wherein customers are selected for service in reverse order of their order of their arrival, i.e., on a last-come, first-served basis.

See

▶ LIFO
▶ Queueing Theory

LCP

Linear complementarity problem.

See

▶ Complementarity Problems
▶ Quadratic Programming

LDU Matrix Decomposition

For a nonsingular square matrix $\boldsymbol{A}$, the transformation by Gaussian elimination of $\boldsymbol{A}$ into the form $\boldsymbol{LDU}$, where $\boldsymbol{L}$ is a lower triangular matrix, $\boldsymbol{D}$ is a diagonal matrix, and $\boldsymbol{U}$ is an upper triangular matrix. It can be written so that the diagonal elements of $\boldsymbol{L}$ and $\boldsymbol{U}$ are equal to one and $\boldsymbol{D}$ is the diagonal matrix of pivots.

See

▶ LU Matrix Decomposition
▶ Matrices and Matrix Algebra

Lean Manufacturing

▶ Quality Control

Lean Six Sigma

▶ Quality Control

Learning

James R. Buck
The University of Iowa, Iowa City, IA, USA

Introduction

Learning is a human phenomenon where performance improves with experience. There are a number of reasons for task improvement. As tasks are repeated, elements of the task are: better remembered, cues are more clearly detected, skills are sharpened, eye-hand coordinations are more tightly coupled, transitions between successive tasks are smoothed, and relationships between task elements are discovered. Barnes and Amrine (1942), Knowles and Bell (1950), Hancock and Foulke (1966), Snoddy (1926), *and* Wickens (1992) have described these and other sources of human performance change. All these causes of individual person improvement manifest themselves in faster performance times, fewer errors, less effort, and there is often a better disposition of the person as a result.

Learning is implied by performance changes due primarily to experience. Changes in the methods of performing a task, replacing human activities with machines, imparting information about the job, training, acquiring performance changes with incentive systems, and many other things can cause performance changes other than learning. Thus, detection involves the identification of an improvement trend as a function of more experience. It also involves the elimination of other explanations for this improvement. Analogous to a theory, learning can never be proved; it can only be disproved.

After detecting learning, measurement and prediction follows. These activities involve fitting mathematical models, called learning curves, to performance data. First, there is the selection of an appropriate model. Following the selection of a model, there is the matter of fitting the selected model to performance data. In some cases alternative models are fit to available data and the quality of fit is a basis in the choice of a model.

Some of those sources which contribute to an individual person's improvement in performance with experience are similar to the causes of improvement by crews, teams, departments, companies, or even industries with experience. As a result, similar terms and descriptions of performance change are often fit to organizational performance changes. However, the term progress curves (Konz 1990) is more often applied to cases involving: assembly lines, crews, teams, departments, and other smaller groups of people, whereas the term experience curves is sometimes applied to larger organizational groups such as companies and industries (Hax and Majluf 1982). A principal distinction between these different types of improvement curves is that between-person activities (e.g., coordination) occur as well as within-person learning. In the case of progress curves, there are improvement effects due to numerous engineering changes. Experience curves also embody scientific and technological improvements, as well progressive engineering changes and individual-person learning. Regardless of the person, persons, or thing which improves or the causes of improvement, the same learning curve models are frequently applied. Progress and experience curves are really forms of personification.

Learning occurs in a number of important applications. One of these applications is the prediction of direct labor changes in production. Not only is this application important to cost estimation, it is also important in production planning and manning decisions. Another application is the selection of an operational method. If there are alternative methods of performing particular operations which are needed, then one significant criterion in the selection of an appropriate method is learning because the average cost can favor one method over another that has lower initial performance costs. In other cases, one operation can cause bottlenecks in others unless the improvements with experience are sufficient over time. Also, production errors can be shown to decrease with experience as another form of learning and so learning is important in quality engineering and control.

Performance Criteria and Experience Units

Performance time is the most common criterion used for learning curves in industry. Production cycles are also the most commonly used variable for denoting experience. If t_i is the performance time on the ith cycle, then a learning curve should predict t_i as a function of n cycles. Since learning implies improvement with experience, then one would expect $t_i \leq t_{i-1}$ for the typical case, $i = 1, 2, \ldots, n$ cycles.

An associated time criterion on the ith cycle is the cumulative average performance time on the ith cycle or A_i. Cumulative average times consists of the sum of all performance times up to and including the nth cycle divided by n. In the first cycle, $A_1 = t_1$. With learning, t_i tends to decrease with i and so does A_i. However, A_i decreases at a slower rate than t_i. This effect can be shown by the first-forward difference of A_i, which is

$$\Delta A_n = A_{n+1} - A_n = \frac{\sum_{i=1}^{n+1} t_i}{n+1} - \frac{\sum_{i=1}^{n} t_i}{n} = \frac{t_{n+1} - A_n}{n+1}. \quad (1)$$

So long as t_{n+1} is less than A_n, then ΔA_n is negative and the cumulative average time continues to decrease. It is also noted in (1) that with sequential values of A_i for $i = 1, 2, \ldots, n$, the values of t_i can be found. On the other hand, A_i can be predicted directly rather than t_i.

Another criterion of interest is accuracy. However, it is usually easier to measure errors in production as the complement of accuracy. Thus, the sequence of production errors are $e_1, e_2, \ldots, e_i, \ldots, e_n$ over n serial cycles where e_i is the number of errors found in a product unit as in typing errors per page (Hutchings and Towill 1975). If the person is doing a single operation on a product unit, then either an error is observed with a unit of production or it is not and observations over a production sequence is a series of zeros and ones. A more understandable practice is to define e_i as the fraction of the possible errors, where the observed number of errors is divided by the m possible

errors at an operation (Fitts 1966; Pew 1969). In this way, e_i is 0, some proper fraction, or 1. It also follows that a learning curve could be fit to the series of e_i values over the n observations sequential units of production or to the cumulative average errors. If learning is present, then one would expect to see a general decrease in e_i with increases in $i = 1, 2, \ldots, n$ and also the cumulative average errors would similarly decrease, but with a rate lag compared to the serial errors.

Pew (1969) invented the speed-accuracy-operating-characteristic graph which provides simultaneous analyses of correlated criteria. This operating characteristic consists of a bivariate graph where one axis denotes performance time per unit (complement is the speed) and the other axis denotes the number of errors per unit (complement is the accuracy). Simultaneous plots of speeds and accuracies with experience would be expected to show increases in both criteria with more experience. The slope of these plots with increases of experience describes bias between these criteria. It should be noted that when the power-form model is used for a prediction of learning performance, then logarithmic axes' measurements will linearize the plots.

Other Learning Metrics

Most applications of learning description, usually known as learning curves, use the production units as experience units, either as single units or lots. The time required to produce that product unit is the corresponding performance units. An alternative approach to predicting learning effects is to describe cumulative time as the experience unit (i.e., hours or days) and the number of production units produced during that experience unit. Thus, for cumulative production time $t = 1, 2, 3, \ldots, k, \ldots, m$ and corresponding production of $n_1, n_2, n_3, \ldots, n_k, \ldots, n_m$. Most learning curve models merely relate n_k to k. An alternative model of learning, which is not often shown, is the discrete exponential model which relates pairs of n_k values as

$$n_k = an_1 + b \tag{2}$$

where a and b are parameters. This model was originally proposed by Pegels (1969) for startup cost prediction. Later, Buck, Tanchoco, and Sweet (1976) showed that this model was really a first-order for-ward-difference equation (Goldberg 1961). It follows in this model that

$$n_k = a^k[n_1 - n^*] + n^* \tag{3}$$

where $n^* = b/(1 - a) > n_1$ and $0 < a < 1$. Since the parameter a is a fraction, the first term of (3) approaches zero with increasing k and so n^* is the asymptote. Accordingly, n_k approaches n^* exponentially with each discrete unit of time. Bevis et al. (1970) provided a similar model as

$$n_k = n^* + [n_1 - n^*]e^{-ck} \tag{4}$$

where k is a continuous measure to time and c is a parameter. Buck and Cheng (1993) used the discrete form in traditional format, but they showed that this model can be more difficult to fit to data than the more common power-form model. It can, however, give a more accurate description of human learning.

See

- Cost Analysis
- Cost-Effectiveness Analysis
- Learning Curves

References

Barnes, R., & Amrine, H. (1942). The effect of practice on various elements used in screw-driver work. *Journal of Applied Psychology, 26*, 197–209.

Bevis, F. W., Finnicat, C., & Towill, D. R. (1970). Prediction of operator performance during learning of repetitive tasks. *International Journal of Production Research, 8*, 293–305.

Buck, J. R., & Cheng, S. W. J. (1993). Instructions and feedback effects on speed and accuracy with different learning curve functions. *IIE Transactions, 25*(6), 34–47.

Buck, J. R., Tanchoco, J. M. A., & Sweet, A. L. (1976). Parameter estimation methods for discrete exponential learning curves. *AIIE Transactions, 8*, 184–194.

Fitts, P. M. (1966). Cognitive aspects of information processing III: Set for speed versus accuracy. *Journal of Experimental Psychology, 71*, 849–857.

Goldberg, S. (1961). *Introduction to difference equations*. New York: Wiley.

Goldberg, M. S., & Touw, A. E. (2003). *Statistical methods for learning curves and cost analysis*. Hannover, MD: INFORMS.
Hancock, W. M., & Foulke, J. A. (1966). Computation of learning curves. *MTM Journal, XL*(3), 5–7.
Hax, A. C., & Majluf, N. S. (1982). Competitive cost dynamics: The experience curve. *Interfaces, 12*(5), 50–61.
Hutchings, B., & Towill, D. R. (1975). An error analysis of the time constraint learning curve model. *International Journal of Production Research, 13*, 105–135.
Knowles, A., & Bell, L. (1950). Learning curves will tell you who's worth training and who isn't. *Factory Management, 108*, 114–115.
Konz, S. (1990). *Work design and industrial ergonomics* (3rd ed.). New York: Wiley.
Pegels, C. C. (1969). On startup of learning curves: An expanded view. *AIIE Transactions, 1*, 216–222.
Pew, R. W. (1969). The speed-accuracy operating characteristic. *Acta Psychologia, 30*, 16–26.
Snoddy, G. S. (1926). Learning and stability. *Journal of Applied Psychology, 10*, 1–36.
Wickens, C. D. (1992). *Engineering psychology and human performance* (2nd ed.). New York: Harper Collins.

Learning Curves

Andrew G. Loerch
Center for Army Analysis, Fort Belvoir, VA, USA

Introduction

With experience and training, individuals and organizations learn to perform tasks more efficiently, reducing the time required to produce a unit of output. This simple and intuitive concept is expressed mathematically through the use of the learning curve.

The learning curve was introduced in the literature by Wright (1936) who observed the learning phenomenon through his study of the construction of aircraft prior to World War II. Since then, these models have been used in the areas of work measurement, job design, capacity planning, and cost estimation in many industries. Yelle (1979) summarized 90 articles dealing with learning curves. Dutton et al. (1984) traced the history of progress functions by examining 300 articles. They note that the terms learning curve, progress function, and experience curve are often used interchangeably. However, many authors differentiate between them in the following way. Learning curves are used to describe only direct-labor learning, while progress functions also incorporate learning by managerial and technical personnel, as well as improvements due to technological change. The term experience curve is used to describe learning or progress at the industry level. Experience curves often use price as a surrogate measure for progress or learning. In the discussion below, no distinctions are made between these terms.

Dutton et al. (1984) also noted that learning curves are frequently confused with economies of scale. Although they are observed together in many cases, the two are separate effects with different causes. Progress and learning can occur in the absence of changes in size or scale of operations.

Basic learning-curve theory is described below, with emphasis given to the so-called power model. Other models are then introduced. Finally, issues regarding the estimation of learning-curve parameters are presented.

The Power Model

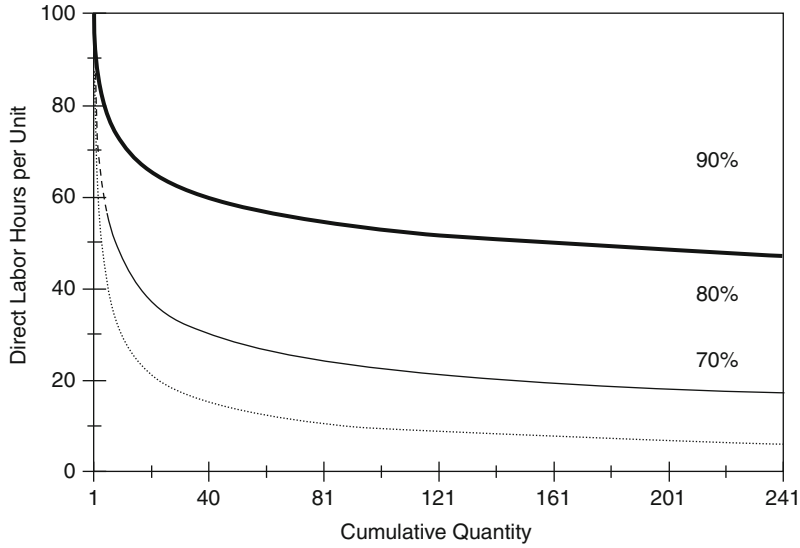
Also known as the log-linear model, the power model is the most frequently encountered implementation of the various learning-curve models. Wright observed that as the quantity of units manufactured doubles, the number of direct labor hours it takes to produce an individual unit decreases at a uniform rate. So, after one doubling of the cumulative production, direct-labor hours may have declined to, say 80% of its previous value. After an additional doubling there is another decline to 80% of that value, or 64% of the original. The learning rate, which is the actual decline per doubling, 80% in the above example, is assumed to be a characteristic of each particular type of manufacturing process.

In this model, learning curves have the following mathematical form:

$$L(y) = Ay^b,$$

where $L(y)$ = the number of hours needed to produce the yth unit, A = the number of hours needed to produce the first unit, y = the cumulative unit number, and b = the learning index, the learning-curve parameter, or the learning-curve slope parameter. To account for the effect of doubling, the learning-curve index is computed as follows:

$$b = (\log r)/(\log 2),$$



Learning Curves, Fig. 1 Learning curves with different rates

where r is the learning rate. Figure 1 shows graphs of three such curves with different learning rates.

Note that this model is also applicable to cost in addition to direct-labor hours. In a cost application, the parameter A would represent the cost of the first unit produced. The use of learning-curve costing is complicated by the problem of accounting for inflation and the change in hourly wages over time. In any event, labor hours can be easily converted into cost.

In the above model, the number of direct-labor hours required to produce the yth unit, or the cost of producing the yth unit is computed. Thus, the model is referred to as the Unit Formulation, and it is attributed to James Crawford who introduced its use to the Lockheed Corporation in 1944 (Smith 1989). A related model based on the original work of Wright is the so-called Cumulative Formulation, where, in the above notation, $L(y)$ would represent the average labor hours or cost of all the units produced through the yth unit. Note that the cumulative formulation tends to smooth the effects of unusually high or low labor hours or costs for individual or groups of units, and it has been found to be more useful for application to batch-type production processes. Although much of the work on learning curves has been directed at specifying the functional relation between unit costs or direct-labor hours and cumulative output, the range of output measures has been expanded to include, for example, industrial accidents per unit output, defects and complaints to quality control per unit output, and service requirements during warranty periods.

Variations of the Power Model

While the log-linear model has been, and is the most widely used model, several other geometries have been found to provide a better fits in particular sets of circumstances. Some of the more well-known models are:

1. Plateau model,
2. Stanford-B model, and
3. S-model.

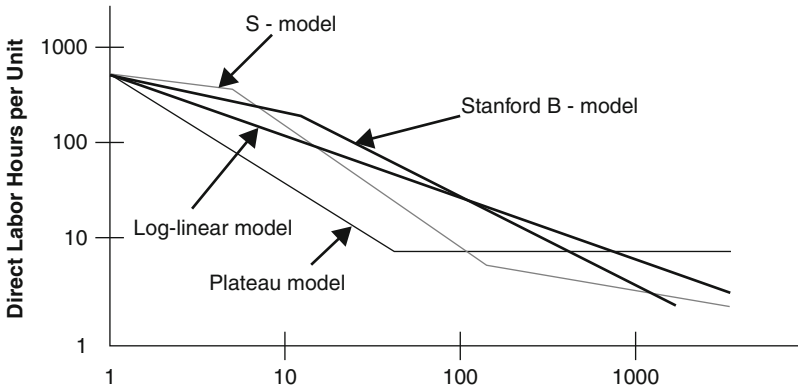
Figure 2 depicts these models on a logarithmic scale.

The plateau model was first described by Conway and Schultz (1959). It is used to represent the phenomenon that the learning phase of a process is finite and is followed by a steady state phase. This model is often associated with machine-intensive manufacturing.

The Stanford-B model, expressed symbolically as

$$L(y) = A(B + y)^b,$$

represents a process that experiences accelerated learning after B units are produced (other notation as



Learning Curves, Fig. 2 Log-log plot of learning curve geometrics

previously defined). This model was developed at the Stanford Research Institute and is useful for processes with design changes (Garg and Milliman 1961).

The *S*-model, described by Cochran (1960), combines reduced learning at the outset of production, with another slackening of learning later in the production process. This model is usually approximated as a three-segment straight line on a log-log graph and is sometimes used for heavy labor-intensive industries.

The choice of the appropriate model is usually based on empirical studies of the process in question and historical experience with similar processes. The utilization of these more complex representations involves increased difficulty in parameter estimation, coupled with limited improvement in accuracy. As such, the basic log-linear model continues to find favor among practitioners.

Other Factors Affecting Learning

Frequently, other factors affect production that, if ignored, could bias the estimation of the rate of learning. As mentioned, the presence of economies of scale would result in the situation where a more than proportional increase in output would be obtained due to an increase in inputs. If the effects of this variable are not controlled for in the estimation of learning rates, and the scale of the operation is gradually increased over time, the amount of learning would be overestimated. Other such factors that are independent of direct labor learning include increased capital investment, multiple shifts, time lapses between performance of operations, and production rate. Argote and Epple (1990) provided a review of the literature regarding the incorporation of factors that affect learning.

Estimation of Learning-Curve Parameters

Most estimation schemes rely on the logarithmic representation of the learning curve, written as follows:

$$\log L = \log A + \log y.$$

The learning-curve parameters, A and b, are estimated either by plotting historical values on a log-log graph and visually fitting a line, or by computing the least squares regression line through the log-log data. Several computer programs are commercially available to estimate the learning-curve parameters.

Frequently, organizations collect historical data for batches or lots, as opposed to discrete units. To estimate the parameters in this case, the batch's average labor or cost and the unit whose labor or cost corresponds to that average, the lot midpoint, must be known. The logarithm of this value is then used as the independent variable in the regression with the log of the average unit cost of the lot as the dependent variable. Note that the unit expressed by the batch size divided by two is not the lot midpoint since the learning curve is nonlinear. The actual lot midpoint, Q, is represented as the following:

$$Q = \left[\frac{(y_l - y_f + 1)(1+b)}{(y_l + .5)^{1+b} - (y_f - .5)^{1+b}}\right]^{-1/b},$$

where $y_f =$ the first unit of the batch, and $y_l =$ the last unit of the batch. Observe that this value cannot be computed without first knowing the learning-curve index, b. As such, the approximate algebraic lot midpoint is used. This value is computed by:

$$Q = \frac{y_f + y_l + 2\sqrt{y_f y_l}}{4}$$

The learning-curve parameters are estimated first using the approximate value of Q for each lot. The value of b is then used to calculate the actual lot midpoint, and the parameters are estimated again, and then iterated until the desired accuracy is obtained.

Concluding Remarks

Research in the area of learning curves has been extensive and many models have been hypothesized to describe the learning process. Learning-curve models have proven to be useful tools in many business and government applications. These include cost estimation, bid preparation and evaluation, labor requirement estimation, establishment of work standards, and financial planning.

See

- ▶ Cost Analysis
- ▶ Cost-Effectiveness Analysis
- ▶ Learning

References

Argote, L., & Epple, D. (1990). Learning curves in manufacturing. *Science, 247*, 920–924.

Buck, J. R., & Cheng, S. W. J. (1993). Instructions and feedback effects on speed and accuracy with different learning curve functions. *IIE Transactions, 25*(6), 34–47.

Cochran, E. B. (1960, July–August). New concepts of the learning curve. *Journal of Industrial Engineering*.

Conway, R. W. & Schultz, A. (1959, January–February). The manufacturing progress function. *Journal of Industrial Engineering*.

Dutton, J. M., Thomas, A., & Butler, J. E. (1984). The history of progress functions as a management technology. *Business History Review, 58*, 1984.

Garg, A., & Milliman, P. (1961). The aircraft progress curve modified for design changes. *Journal of Industrial Engineering, 12*, 23.

Goldberg, M. S., & Touw, A. E. (2003). *Statistical methods for learning curves and cost analysis*. Hanover, MD: INFORMS.

Smith, J. (1989). *Learning curves for cost control, industrial engineering and management press*. Norcross, GA: Institute of Industrial Engineers.

Wright, T. P. (1936). Factors affecting the cost of airplanes. *Journal of Aeronautical Sciences, 3*(4), 122–128.

Yelle, L. E. (1979). The learning curve: Historical review and comprehensive survey. *Decision Sciences, 10*, 302–328.

Least-Squares Analysis

- ▶ Quadratic Programming
- ▶ Regression Analysis

Leontief Matrix

- ▶ Input–Output Analysis

Level Crossing Methods

Percy H. Brill
University of Windsor, Windsor, Ontario, Canada

Introduction

Level crossing methods for obtaining probability distributions in stochastic models such as queues and inventories were originated by Brill (1975, 1976, 1979) and elucidated further in Brill and Posner (1974, 1975, 1977, 1981), and Cohen (1976, 1977). These methods began as an essential part of system point theory and are also known as system point analysis, sample path analysis, or level crossing technique, approach, theory, or analysis in the literature (Brill 1975, 2008). Level crossing methods are very useful rate conservation techniques for stochastic models (Miyazawa 1994).

Model and Stationary Distribution

Consider a stochastic process $\{W(t), t \geq 0\}$ where both the parameter set and state space are continuous. The random variable $W(t)$ at time point t may denote the content of a dam with general efflux, the stock on hand in an (s,S) or (r,nQ) inventory system with stock decay, or the virtual wait or workload in a queue. Assume that upward jumps of $\{W(t)\}$ occur at Poisson rate λ_u and downward jumps at Poisson rate λ_d. Let upward and downward jump magnitudes have cumulative distribution function (CDF) B_u and B_d, respectively. Assume that the model parameters are such that the stationary distribution of $W(t)$ exists as $t \to \infty$. Let G and g denote the stationary CDF and probability density function (PDF), respectively. The aim here is to obtain expressions for g and G in terms of the model parameters by using a level crossing approach.

Sample Paths

A sample path of the $\{W(t)\}$ process is a right-continuous, real-valued function on the nonnegative reals whose value at time-point t is the realized value of random variable $W(t)$. Denote an arbitrary sample path by the function $X(t)$, $t \geq 0$. The function X has either jump or removable discontinuities on a sequence of strictly increasing time points $\{\tau_n, n = 0, 1, \ldots\}$, where $\tau_0 = 0$ without loss of generality. Typically, the time points $\{\tau_n\}$ represent input or output epochs in dams, arrival epochs in queues, or demand or replenishment epochs in inventories. Assume that when a sample path is positive valued, it decreases continuously on time segments between jump points, described by $dX(t)/dt = -rX(t), X(t) > 0, \tau_n \leq t < \tau_{n+1}$, $n = 0, 1, 2, \ldots$ wherever the derivative exists, and where $r(x) > 0$ for $x > 0$. Note that for the virtual wait process in queues, $r(x) = 1(x > 0)$ and $r(0) = 0$. In an (s,S) continuous review inventory system, where the stock on hand decays at constant rate k, then $r(x) = k$ for all x between the reorder level s and order-up-to-level S.

Level Crossing by Sample Paths

Let x denote a fixed state space level and t_0 an arbitrary positive time point. Let t_0 be one of the jump time points $\{\tau_n\}$, $n = 1, 2, \ldots$ and let d_0 and u_0 denote the corresponding downward and upward jump magnitudes, respectively, where at least one of u_0, d_0, is strictly positive. The sample path may down cross level x at $t_0 > 0$ if t_0 is any positive epoch, but it can up cross level x at t_0 only if t_0 is one of the $\{\tau_n\}$.

If a sample path down crosses level x at t_0 which is not one of the $\{\tau_n\}$, then the down crossing is a continuous down crossing, since the sample path is continuous at t_0. If a sample path down crosses level x at t_0 which is one of the $\{\tau_n\}$, then the downward jump of magnitude d_0 brings it from above x to a level below x. If a sample path up crosses level x at t_0, then, necessarily, t_0 is one of the epochs $\{\tau_n\}$, and the upward jump of magnitude u_0 brings it from below x to a level above x.

If both u_0 and d_0 are strictly positive at t_0 which is one of the $\{\tau_n\}$, the model mechanism would determine whether the downward or upward jump is considered to precede the other. In inventories without lead time, for example, stock depletions due to demands (downward jumps) precede stock replenishments (upward jumps). The jumps are not part of the sample path per se, but serve only to construct the path. One may also define level crossings at some time point t_0 by considering the net jump which has magnitude $|u_0 - d_0|$ and upward (downward) direction if $u_0 > d_0 (u_0 < d_0)$.

Level Crossings and the Stationary Distribution

Down crossings — Let $D^u_{ct}(x)$ denote the number of continuous down crossings of level x and $D^j_t(x)$, the number of jump down crossings of level x during $(0, t)$, $t > 0$. Then, for $r(x) = 1, x > 0$ and $r(0) = 0$, it follows with probability 1 that

$$\lim_{t\to\infty} \frac{D^c_t(x)}{t} = r(x)g(x) \quad (\text{for all } x), \tag{1}$$

(Brill 1975). The following also holds with probability 1:

$$\lim_{t\to\infty} \frac{D^j_t(x)}{t} = \lambda_d \int_{y=x}^{\infty} \bar{B}_d(y-x) g_d(y) dy \quad (\text{for all } x), \tag{2}$$

where g_{od} is the limiting PDF at embedded downward jump points as $t \to \infty$ and $\bar{B} \equiv 1 - B$.

Both Eqs. 1 and 2 also hold upon replacing $D^u_{ct}(x)$ and $D^u_{ty}(x)$ by their expectations, denoted by $E[D^c_t(x)]$ and $E[D^u_{ty}(x)]$, respectively, and deleting with probability 1. For exponentially distributed interarrivals between downward jumps (Poisson downward jumps), then $g_d \equiv g$, which is the *PASTA* principle.

Up crossings —Let $U^j_t(x)$ denote the number of jump up crossings of level x during $(0, t)$. Then, with probability 1,

$$\lim_{t\to\infty} \frac{U^j_t(x)}{t} = \lambda_u \int_{-\infty}^{x} \bar{B}_u(x-y) g_u(y) dy \qquad (\text{for all x}), \tag{3}$$

where g_u is the limiting PDF at embedded upward-jump time points as $t \to \infty$ (Brill 1975).

Formula (3) gives an expression for the long-run up crossing rate of level x by any typical sample path at upward jump points, in terms of an integral of the density g_u. For Poisson upward jumps, $g_u \equiv g$ by the PASTA principle.

A Conservation Law for Level Crossings

For each state space level, the following conservation law holds:

long run total down crossing rate = long run total up crossing rate.

This conservation law, together with Eqs. 1, 2 and 3, enables one to write an integral equation for the PDF g in which every term has a precise interpretation as a sample-path down or up crossing rate, namely,

$$r(x)g(x) + \lambda_d \int_{y=x}^{\infty} \bar{B}_d(y-x) g(y) dy = \lambda_u \int_{y=-\infty}^{x} \bar{B}_u(x-y) g(y) dy \qquad (\text{for all } x). \tag{4}$$

In (4), the left-hand side depicts the total sample path long-run down crossing rate of level x, while the right-hand side depicts the long-run up crossing rate of the level x. Equation (4) is then solved for g by using standard applied mathematics techniques.

Applicability

The level crossing technique is applicable to dams with limited capacity, blocked-input rules, various control level policies, etc.; to complex variants of M/G/1, M/M/c, G/M/1 queues with reneging, bounded virtual wait, server vacations, various state dependencies, cyclic-service queues; and to a wide class of inventory, production/inventory, counter, risk reserve, and related models.

The same level crossing ideas as in Eqs. 1, 2 and 3 have been applied to cycles in regenerative processes by Cohen (1976, 1977). Upon combining the regenerative-processes level crossing approach and the embedded level crossing technique of Brill (1976, 1979) with the previously widely known bubble diagram method (rate into a state = rate out of that state) for discrete state continuous time Markov chains, level crossing methods can be applied to obtain probability distributions and other characteristics in a broad class of stochastic models.

Level Crossing Estimation

The principle established in formula (1) motivates the idea of using $D^c_t(x)/[tr(x)]$ as an estimate for $g(x)$ when t is large. Level crossing estimation (also known as system point estimation) consists of three main steps: (I) simulating a single sample path over a large simulated time t; (ii) enumerating the continuous down crossings of all state space levels over $(0, t)$; and (iii) computing both point and interval estimates of g, G and the moments (Brill 1991).

See

- Inventory Modeling
- Markov Processes
- PASTA
- Queueing Theory

References

Azoury, K., & Brill, P. H. (1986). An application of the system-point method to inventory models under continuous review. *Journal of Applied Probability, 23*, 778–789.

Brill, P. H. (1975). System point theory in exponential queues. (*Ph.D. Dissertation*, University of Toronto).

Brill, P. H. (1976). Embedded level crossing processes in dams and queues. WP #76-022, Department of Industrial Engineering, University of Toronto.
Brill, P. H. (1979). An embedded level crossing technique for dams and queues. *Journal of Applied Probability, 16*, 174–186.
Brill, P. H. (1991). Estimation of stationary distributions in storage processes using level crossing theory. Proceedings of the Statistical Computing Section, American Statistical Association, 172–177.
Brill, P. H. (2008). *Level crossing methods in stochastic models*. New York: Springer.
Brill, P. H., & Posner, M. J. M. (1974). On the equilibrium waiting time distribution for a class of exponential queues. WP #74-012, Department of Industrial Engineering, University of Toronto.
Brill, P. H., & Posner, M. J. M. (1975). Level crossings in point processes applied to queues. WP #75-009, Department of Industrial Engineering, University of Toronto.
Brill, P. H., & Posner, M. J. M. (1977). Level crossings in point processes applied to queues: Single server case. *Operations Research, 25*, 662–673.
Brill, P. H., & Posner, M. J. M. (1981). The system point method in exponential queues: A level crossing approach. *Mathematics of Operations Research, 6*, 31–49.
Cohen, J. W. (1976). *On regenerative processes in queueing theory* (Lecture notes in economics and mathematical systems, p. 121). New York: Springer-Verlag.
Cohen, J. W. (1977). On up and down crossings. *Journal of Applied Probability, 14*, 405–410.
Miyazawa, M. (1994). Rate conservation laws: A survey. *Queueing Systems: Theory & Applications, 18*, 1–58.
Ross, S. (1985). *Introduction to probability models* (4th ed.). New York: Academic Press.

Level Curve

Also called isovalue contour: a curve along which the values of a given associated function remain constant.

See

► Isoquant

Lexicographic Ordering

An ordering of a set of vectors based on the lexicopositive (negative) properties of the vectors. For example, the sequence of vectors $\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_q\}$ is ordered in a lexicographic sense if $\boldsymbol{x}_i - \boldsymbol{x}_j$ is lexico-positive for $i > j$. Such orderings are similar to dictionary ordering of words and are used to prove finiteness of the simplex algorithm.

See

► Cycling
► Lexico-Positive (Negative) Vector

Lexico-Positive (Negative) Vector

A vector $\boldsymbol{x} = (x_1,\ldots, x_n)$ is called lexico-positive (negative) if $\boldsymbol{x} \neq 0$ and the first nonzero term is positive (negative). The vector $\boldsymbol{x}$ is lexico-negative if $-\boldsymbol{x}$ is lexico-positive. A vector $\boldsymbol{x}$ is greater than a vector $\boldsymbol{y}$ in a lexico-positive sense if $\boldsymbol{x} - \boldsymbol{y}$ is lexico-positive.

See

► Lexicographic Ordering

LGP

Linear goal programming.

See

► Goal Programming

Libraries

Arnold Reisman[1] and Xiaomei Xu[2]
[1]Reisman and Associates, Shaker Heights, OH, USA
[2]Cleveland, OH, USA

The American Heritage Dictionary of the English Language (1976, p. 753) defines a library is

"a repository for literary and artistic materials such as books, periodicals, newspapers, pamphlets, and prints kept for reading or reference." This rather classical notion of a library does not recognize the fact that libraries are now a subset of the broader field known as Information Systems (IS). Nevertheless, the scope of this article will be delimited to institutions which can be defined as above, albeit with some leeway.

The history of the application of operations research/management science to libraries is not very distinguished. Contributions in the library field were constrained up to and through the decade of the 1970s by the fact that few operations researchers chose libraries as a field of interest. Moreover, librarians have not sought out operations researchers to help in their problem solving, nor did they offer a particularly fertile environment for doing OR studies (Chen 1974). On the other hand, since the 1970s, computer science has made significant inroads into the library field by merging with library science to create local and extended area computer networks linking users with comprehensive databases.

The first known application of OR to libraries in the United States can be credited to Bacon and Machol (1958). The 1960s recorded a more widespread interest (Cox 1964; Morse 1968; Cook 1968). A comprehensive review on library operations research was done by Kantor (1979). In that review, Kantor summarized all of the previous review articles. Most noteworthy of these from the OR point of view are the bibliographies by Slamecka (1972) and Kraft and McDonald (1977), and surveys and/or assessments by Bommer (1975), Kraft and McDonald (1976), Leimkuhler (1970, 1972, 1977a, 1977b), Churchman (1972) and Morse (1972).

Literature on utilization of OR in libraries has classified the field in several different ways. Kantor (1979) classified papers and projects into the following groups according to the purpose of the research: system description; modeling the system; parameter identification; optimization or multi-valuation; and application. Rowley and Rowley (1981) classified the work by the nature of the research (recurrent problems, on/off decisions, etc.). For the purposes of this article, a three-dimensional classification is used with one of the dimensions adopting Rowley's (1981) classification, with slight modifications. Based on the type of problems being analyzed, the application areas are operational or recurrent problems, such as book storage problems; strategies or on/off decisions, such as library location problems; and control/design problems, such as loan policy problems (Rowley and Rowley 1981).

The second dimension on the application of OR in libraries is a classification according to the type of OR techniques used:

1. *Queueing models* – Given the average book circulation time ($1/\mu$) and the mean number of persons who borrow the book (λ), the expected circulation rate of that particular book is derived using queueing theory (Morse 1968).
2. *Simulation* – With the number of staff, the volumes of various jobs (users' requests, new issues, overdue fees, etc.) and the job processing times specified, simulation is used to estimate the delays, processing times and utilization of each member of staff and the whole facility (Thomas and Robertson 1975).
3. *Facility location algorithms* – The library facilities and relocation problems are discussed by Min (1988).
4. *Mathematical programming* – If there are two types of information services, both of which share the same set of resources (staff time in scanning, indexing, abstracting, etc.), and each of them has a different unit profit, a linear programming problem is used to find out how many services of each type to produce to maximize the total profit (Rowley and Rowley 1981, *58–64*).
5. *Network flow models* – Given the heights and thicknesses of a given collection of books and the cost of different shelf heights, a network model is developed to determine the optimal number of shelf heights for minimizing shelving costs through finding the shortest path in a directed network (Gupta and Ravindram 1974).
6. *Decision theory* – A decision regarding whether or not to install a library security system is addressed given the installation cost and the probabilities of success and failure (Rowley and Rowley 1981, 91–92).
7. *Search theory* – Patterns of browsing in libraries are addressed in Morse (1970).
8. *Transportation models* – A routing problem is explored for a vehicle delivering materials to branches (Heinritz and Hsiao 1969; McClure 1977).

9. *Inventory control theory* – An EOQ model is used to determine the optimal order quantity for the stock of a certain library supply (Rowley and Rowley 1981, 111–116).
10. *Probability and statistics* – Library book circulation and individual book popularities are considered as probabilistic processes by Gelman and Sichel (1987) who demonstrated the superiority of beta over the negative binomial distribution.
11. *Benefit cost analysis* – Library planning is addressed by Leimkuhler and Cooper (1971).

Each of these categories could be, in turn, further characterized by whether or not the research work was grounded, e.g., based on real world library systems involving real data and/or bona fide librarians in the study as opposed to models which were basically what might be called logico/deductive. A more thorough discussion is given in Reisman and Xu (1994), where Table I, page 37, provides a taxonomic review of the vast bulk of the literature in the field.

As can be seen from the above delineation and the referenced table, the utilization of OR in libraries is far from achieving its full potential. Except for simulation and probability and statistics based applications, the bulk of the literature is not well grounded in real life settings. The literature reflects the gap between the complex mathematical models in OR and the usually not very quantitatively educated library workers (Stueart and Moran 1987). To enhance the application of OR in libraries, Bommer (1975) suggested a closer working relationship between operations researchers and library managers.

See

► Information Systems and Database Design in OR/MS

References

Bacon, F. R. Jr., & Machol, R. E. (1958). *Feasibility analysis and use of remote access to library card catalogs*. Paper, presented at the Fall meeting of ORSA (Unpublished).

Bacon, F. R., Jr., Churchill, N. C., Lucas, C. J., Maxfield, D. K., Orwant, C. J., & Wilson, R. C. (1958). *Applications of a teller reference system to divisional library card catalogues: A feasibility analysis*. Ann Arbor, MI: Engineering Research Institute, University of Michigan.

Bommer, M. (1975). Operations research in libraries: A critical assessment. *Journal of the American Society for Information Science, 26*, 137–139.

Chen, Ching-chih. (1974). *Applications of operations research models to libraries: A case study of the use of monographs in the Francis A. Countway Library of Medicine, Harvard University*. Unpublished Ph.D. dissertation, Case Western Reserve University, School of Library Science, Cleveland, OH.

Churchman, C. W. (1972). Operations research prospects for libraries: The realities and ideals. *Library Quarterly, 42*, 6–14.

Cook, J. J. (1968). Increased seating in the undergraduate library: A study in effective space utilization. In B. R. Burkhalter (Ed.), *Case studies in systems analysis in a university library* (pp. 142–170). Metuchen, NJ: Scarecrow Press.

Cox, J. G. (1964). *Optimal storage of library material*. Unpublished Ph.D. dissertation, Purdue University Libraries, Lafayette, Indiana.

Gelman, E., & Sichel, H. S. (1987). Library book circulation and the beta-binomial distribution. *Journal of the American Society for Information Science, 38*, 4–12.

Gupta, S. M., & Ravindram, A. (1974). Optimal storage of books by size: An operations research approach. *Journal of the American Society for Information Science, 25*, 354–357.

Heinritz, F. J., & Hsiao, J. C. (1969). Optimum distribution of centrally processed material. *Library Resources and Technical Services, 13*, 206–208.

Kantor, P. (1979). Review of library operations research. *Library Research, 1*, 295–345.

Kraft, D. H., & McDonald, D. D. (1976). Library operations research: Its past and our future. In D. P. Hammer (Ed.), *The information age* (pp. 122–144). Metuchen, NJ: Scarecrow Press.

Kraft, D. H., & McDonald, D. D. (1977). Library operations research: A bibliography and commentary of the literature. *Information, Reports and Bibliographies, 6*, 2–10.

Leimkuhler, F. F. (1970). Library operations research: An engineering approach to information problems. *Engineering Education, 60*, 363–365.

Leimkuhler, F. F. (1972). Library operations research: A process of discovery and justification. *Library Quarterly, 42*, 84–96.

Leimkuhler, F. F. (1977a). Operational analysis of library systems. *Information Processing and Management, 13*, 79–93.

Leimkuhler, F. F. (1977b). Operations research and systems analysis. In F. W. Lancaster & C. W. Cleverdon (Eds.), *Evaluation and scientific management of libraries and information centres* (pp. 131–163). Leyden, The Netherlands: Nordhoff.

Leimkuhler, F. F., & Cooper, M. D. (1971). Analytical models for library planning. *Journal of the American Society for Information Science, 22*, 390–398.

McClure, C. R. (1977). Linear programming and library delivery systems. *Library Resources and Technical Services, 21*, 333–344.

Min, H. (1988). The dynamic expansion and relocation of capacitated public facilities: A multi-objective approach. *Computers and Operations Research (UK), 15*, 243–252.
Morse, P. M. (1968). *Library effectiveness: A systems approach*. Cambridge, MA: MIT Press.
Morse, P. M. (1970). Search theory and Browsing. *Library Quarterly, 40*, 391–408.
Morse, P. M. (1972). Measures of library effectiveness. *Library Quarterly, 42*, 15–30.
Reisman, A., & Xu, X. (1994). Operations research in libraries: A review of 25 years of activity. *Operations Research, 42*, 34–40.
Rowley, J. E., & Rowley, P. J. (1981). *Operations research: A tool for library management* (pp. 3–4). Chicago: American Library Association.
Slamecka, V. (1972). A selective bibliography on library operations research. *Library Quarterly, 42*, 152–158.
Stueart, R. D., & Moran, B. B. (1987). *Library management* (3rd ed., pp. 200–202). Littleton, CO: Libraries Unlimited.
Thomas, P. A., & Robertson, S. E. (1975). A computer simulation model of library operations. *Journal of Documentation, 31*, 1–16.

LIFO

The Last-In, First-Out queue discipline in which customers are selected for service in reverse order of their arrival (meant to be equivalent to the last-come, first-served scheme).

See

- ▶ LCFS
- ▶ Queueing Theory

Light-Tailed Distribution

A probability distribution that has an exponentially decaying complementary CDF, e.g., the normal (Gaussian) and exponential distributions.

See

- ▶ Heavy-Tailed Distribution

Likelihood Ratio Method

A method for gradient estimation in simulation used for sensitivity analysis and optimization; also known as the score function method.

See

- ▶ Perturbation Analysis
- ▶ Score Functions
- ▶ Simulation Optimization

Limiting Distribution

Let $p_{ij}(t)$ be the probability that a stochastic process takes on value j at time t (discrete or continuous), given that it began at time 0 from state i. If for each j, $p_{ij}(t)$ approaches a limit p_j as $t \to \infty$ independent of i, the set $\{p_j\}$ is called the limiting or steady-state distribution of the process. For Markov chains in discrete time, the existence of a limiting distribution implies that there is a stationary (or invariant) distribution found from $\boldsymbol{\pi} = \boldsymbol{\pi P}$, where $\boldsymbol{P}$ is the single-step transition matrix, such that $\boldsymbol{\pi} = \boldsymbol{p}$. Similarly, for continuous-time chains, the steady-state distribution is the probability vector satisfying the global balance equations $\boldsymbol{\pi Q} = \boldsymbol{0}$, where $\boldsymbol{Q}$ is the transition rate matrix.

See

- ▶ Markov Chains
- ▶ Markov Processes
- ▶ Stationary Distribution
- ▶ Statistical Equilibrium

Lindley's Equation

An integral equation for the steady-state waiting-time distribution in the first-come, first-served, single-server G/G/1 queue. If $W_q(x)$, $x \geq 0$, is the

steady-state distribution function of the delay or waiting time in the queue, then, for $x \geq 0$,

$$W_q(x) = \int_{-\infty}^{x} W_q(x-y)dU(y)$$

with $W_q(x) = 0$ for $x < 0$, where the function $U(y)$ is the distribution function of the random variable defined as the service time minus the interarrival time.

Lindley's equation can also be used to refer to the finite-time transient recursive equation relating delays in the first-come, first-served, single-server G/G/1 queue as follows:

$$D_{n+1} = \max(0, D_n + S_n - A_n),$$

where D_n is the delay of the nth arriving customer, S_n is the service time of the nth arriving customer, and A_n is the interarrival time between the nth and $(n + 1)$st arriving customer.

See

- ▶ Kendall's Notation
- ▶ Queueing Theory

Line

A line is the set of points $\{x|x = (1-\lambda)x_1 + \lambda x_2\}$, where x_1 and x_2 are points in n-dimensional space and λ is a real number. The line passes through the points x_1 and x_2, $x_1 \neq x_2$.

Line Segment

The straight line joining any two points in n-dimensional real space is a line segment. More specifically, if x_1 and x_2 are the two points, then the set of points $\{x|x = (1-\lambda)x_1 + \lambda x_2, 0 \leq \lambda \leq 1\}$ is the line segment joining x_1 and x_2.

See

- ▶ Line

Linear Combination

For a set of vectors $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$, a linear combination is another vector $\boldsymbol{y} = \sum_j \alpha_j \boldsymbol{x}_j$, where the scalar coefficients α_j can take on any values.

Linear Equation

The mathematical form $a_1x_1 + a_2x_2 + \ldots + a_nx_n = b$ is a linear equation, where the a_j and b can take on any values.

See

- ▶ Hyperplane

Linear Functional

A linear functional $f(\boldsymbol{x})$ is a real-valued function defined on an n-dimensional vector space such that, for every vector $\boldsymbol{x} = \alpha\boldsymbol{u} + \beta\boldsymbol{v}$, $f(\boldsymbol{x}) = f(\alpha\boldsymbol{u} + \beta\boldsymbol{v}) = \alpha f(\boldsymbol{u}) + \beta f(\boldsymbol{v})$ for all n-dimensional vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ and all scalars α and β.

Linear Inequality

The mathematical form $a_1x_1 + a_2x_2 + \ldots + a_nx_n \leq b$ or $a_1x_1 + a_2x_2 + \ldots + a_nx_n \geq b$ is a linear inequality, where the numbers a_j and b can take on any values. The set of vectors $\boldsymbol{x} = (x_1, \ldots, x_n)$ that satisfy the inequality form a solution half space.

See

- ▶ Hyperplane

Linear Programming

Frederick S. Hillier
Stanford University, Stanford, CA, USA

Introduction

Linear programming is one of the most widely used techniques of operations research and management science. Its name means that planning (programming) is being done with a mathematical model (called a linear-programming model) where all the functions in the model are linear functions.

Linear Programming Models

Linear programming models come in a variety of forms. To illustrate one common form, consider the problem of determining the most profitable mix of products for a manufacturer. Let n be the number of possible products. For each product j ($j = 1, 2, \ldots, n$), a decision variable x_j is introduced to represent the decision on its production rate. Let c_j be the profit per unit of product j produced, and let Z be the total rate of profit resulting from the choice of product mix. This choice is constrained by the limited capacities of the production facilities available for these products. Let m be the number of different types of facilities needed. For each type i ($i = 1, 2, \ldots, m$), let b_i be the amount of capacity available per unit time and let a_{ij} be the amount of capacity used by each unit produced of product j ($j = 1, 2, \ldots, n$). The resulting linear programming model then is to choose $x_1, x_2, \ldots, x_n$ so as to

$$\begin{aligned} \text{Maximize } Z &= c_1x_1 + c_2x_2 + \cdots + c_nx_n \\ \text{subject to : } a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &\le b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &\le b_2 \\ \vdots \qquad\qquad & \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &\le b_m \end{aligned}$$

and

$$x_1 \ge 0, x_2 \ge 0, \ldots, x_n \ge 0.$$

The linear function being maximized in this model is called the objective function. The m inequalities with a linear function on the left-hand side are referred to as functional constraints (or structural constraints), and the inequalities in the bottom row are nonnegativity constraints. The constants (c_j, b_i, and a_{ij}) are the parameters of the model. Any choice of values of ($x_1, x_2, \ldots, x_n$) is called a solution, whereas a solution satisfying all the constraints is a feasible solution, and a feasible solution that maximizes the objective function is an optimal solution.

Many other applications of linear programming having nothing to do with product mix also fit this same form for the model. In these cases, activities of some other kind replace the production of products, and resources of some other kind replace production facilities. For each activity j ($j = 1,2, \ldots, m$), the decision variable x_j represents the decision on the level of that activity. The problem then is to allocate these limited resources to these interrelated activities so as to obtain the best mix of activities (i.e., an optimal solution) according to the overall measure of performance adopted for the objective function.

Another common form for a linear programming model is to minimize the objective function, subject to functional constraints with $\ge$ signs and nonnegativity constraints. A typical interpretation then is that the objective function represents the total cost for the chosen mix of activities and the functional constraints involve different kinds of benefits. In particular, the function on the left-hand side of each functional constraint gives the level of a particular kind of benefit that is obtained from the mix of activities, and the constant on the right-hand represents the minimum acceptable level for that benefit. The problem then is to determine the mix of activities that gives the best tradeoff between cost and benefits according to the model.

Still other linear-programming models have an equality instead of inequality sign in some or all of the functional constraints. Such constraints represent fixed requirements for the value of the function on the left-hand side.

It is fairly common for large linear-programming models to include a mixture of functional constraints — some with $\le$ signs, some with $\ge$ signs, and some with $=$ signs. Nonnegativity constraints always have a $\ge$ sign, but it occasionally is appropriate to delete this kind of constraint for some or all of the decision variables.

Successful applications of linear programming sometimes use very large models. As described in a later section, exceptionally efficient algorithms are available for solving these models. When using state-of-the-art implementations of these algorithms and a powerful desktop computer or workstation, a model with several thousand functional constraints and decision variables is considered to be of moderate size. Having a few tens of thousands of functional constraints and even more decision variables is not considered particularly large. Far bigger problems with millions of functional constraints and decision variables sometimes are solved, depending largely on whether they have a special structure that can be exploited.

With large models, it is inevitable that mistakes and faulty decisions will be made initially in formulating the model and inputting it into the computer. Therefore, a thorough process of testing and refining the model, i.e., model validation, is needed. The usual end-product is not a single static model, but rather a long series of variations on a basic model to examine different scenarios as part of post-optimality analysis (discussed later). A sophisticated modeling language usually is needed to efficiently formulate the model and then to expedite a number of model management tasks, including accessing data, transforming data into model parameters, modifying the model whenever desired, and analyzing solutions from the model.

Some Applications of Linear Programming

The applications of linear programming have been remarkably diverse. They all involve determining the best mix of activities, where the decision variables represent the levels of the respective activities, but these activities arise in a wide variety of contexts. In the context of financial planning, the activities might be investing in individual stocks and bonds (portfolio selection), or undertaking capital projects (capital budgeting), or drawing on sources for generating working capital (financial-mix strategy). In the context of marketing analysis, the activities might be using individual types of advertising media, or performing marketing research in segments of the market. In the context of production planning, applications range widely from the product-mix problem (discussed earlier) to the blending problem (determining the best mix of ingredients for various individual final products), and from production scheduling to personnel scheduling.

In addition to manufacturing, these kinds of production planning applications also arise in agricultural planning, health-care management, the planning of military operations, policy development for the use of natural resources, etc.

Linear programming has had a great impact on improving the efficiency and profitability of numerous organizations around the world. A considerable number of these applications have won a prestigious prize in the annual international competition for the Franz Edelman Award for Achievement in Operations Research and the Management Sciences. To mention a few typical award-winning applications: Bixby et al. (2006) describe how Swift & Company saved $12 million in 1 year by optimizing its product mix while dynamically scheduling its beef-fabrication operations at five plants in real time as it receives orders; Lee and Zaider (2008) discuss how a breakthrough in optimizing the application of brachytherapy to prostrate cancer is having a profound impact on both health care costs (potentially saving $500 million annually) and quality of life for treated patients; Holloran and Bryne (1986) were early pioneers in applying linear programming at United Airlines to design the work schedules for all the employees at the various reservation offices and airports, thereby saving the company more than $6 million annually; Leachman, Kang, and Lin (2002) describe how Samsung Electronics Corp. captured an additional $200 million in annual sales revenue by using a linear- programming model with tens of thousands of decision variables and functional constraints to increase the efficiency of its processes for manufacturing random access memory devices. Hillier and Lieberman (2010, Chap. 3) also reference other award-winning applications of linear programming.

Another important kind of application of linear programming arises from its close relationship to several other important areas of operations research and management science, including integer programming, nonlinear programming, and game theory. Linear programming often is useful to help solve problems in these other areas as well.

Some Special Types of Linear Programming Models

One particularly important special type of linear programming problem is the transportation problem. A typical application of the transportation problem is to determine how a corporation should distribute a product from its various factories to various distributors. In particular, given the amount of the product produced at each factory and the amount needed by each distributor, one can determine how much to ship from each factory to each distributor in order to minimize total shipping cost. Other applications extend to areas such as production scheduling.

Camm et al. (1997) describe an award-winning application of the transportation problem at Procter & Gamble that saved over $200 million annually by redesigning the company's production and distribution system for its North American operations. A major part of the study revolved around formulating and solving transportation problems for individual product categories.

The assignment problem is a special type of linear-programming problem where assignees are being assigned to perform tasks. For example, the assignees might be employees who need to be given work assignments. Assigning people to jobs is a common application of the assignment problem. However, the assignees need not be people. They also could be machines, or vehicles, or plants, or even time slots to be assigned tasks. It can be shown that the mathematical structure of the model for the assignment problem is a special case of that for the transportation problem.

Both the transportation problem and the assignment problem are a special case of another key type of linear-programming problem, called the minimum-cost network-flow problem, that involves determining how to distribute goods through a distribution network at a minimum total cost. In particular, the nodes of this network include at least one supply node and at least one demand node, and then the rest of the nodes are transshipment nodes. Given the capacity of each arc for transmitting flow, the objective is to minimize the total cost of sending the supply from the supply nodes through the network to satisfy the given demand at the demand nodes.

Klingman et al. (1987) describe a classic award-winning application of this type at the Citgo Petroleum Corporation. This minimum-cost network-flow problem involved the distribution of petroleum products through a distribution network consisting of pipelines, tankers, barges, and hundreds of terminals. This application is credited with saving the company well over $15 million annually. (Another application of linear programming involving Citgo's refinery operations was implemented at about the same time and achieved additional savings of about $50 million per year).

Another special case of the minimum-cost network-flow problem is the maximum-flow problem. Given a connected network with capacity constraints on the maximum flow through each arc, the objective now is to maximize the flow through the network from the source node to the sink node. Some typical applications include maximizing the flow through a distribution network, or through a supply network, or through a system of pipelines, or through a system of aqueducts, or through a transportation network.

The shortest-path problem (also called the shortest-route problem) is still another important special type of linear-programming problem that is also a special case of the minimum-cost network-flow problem. The objective now is to find the path through a network from an origin to a destination that minimizes the total distance traveled. Arc distances also can represent costs or times so the objective becomes to minimize the total cost or total time of a sequence of activities.

Ireland et al. (2004) describe how the Canadian Pacific Railway saves roughly $100 million annually by using network optimization techniques to route its freight each day over a massive rail network that encompasses much of North America. Numerous shortest-path problems are solved each day as part of the overall approach for this award-winning application.

There have been many other award-winning applications of the special types of linear-programming problems that are described above. Hillier and Lieberman (2010, Chap. 9) reference some of these applications.

Solving Linear Programming Models

Two crucial events have been primarily responsible for the great impact of linear programming since its

emergence in the middle of the twentieth century. One was the invention in 1947 by George Dantzig of a remarkably efficient algorithm, called the simplex method, for finding an optimal solution for a linear-programming model. The second crucial event was the computer revolution that makes it possible for the simplex method to solve huge problems.

The simplex method exploits some basic properties of optimal solutions for linear programming models. Because all the functions in the model are linear functions, the set of feasible solutions (called the feasible region) is a convex polyhedral set. The vertices (extreme points) of the feasible region play a special role in finding an optimal solution. A model will have an optimal solution if it has any feasible solutions (all the constraints can be satisfied simultaneously) and the constraints prevent improving the value of the objective function indefinitely. Any such model must have either exactly one optimal solution or an infinite number of them. In the former case, the one optimal solution must be a vertex of the feasible region. In the latter case, at least two vertices must be optimal solutions, and then all convex-linear combinations of these vertices also are optimal. It is sufficient, therefore, to find the vertices with the most favorable value of the objective function in order to identify all optimal solutions.

Based on these facts, the simplex method is an iterative algorithm that only examines vertices of the feasible region. At each iteration, it uses algebraic procedures to move along an outside edge of the feasible region from the current vertex to an adjacent vertex that is better. The algorithm terminates (except perhaps for checking ties) when a vertex is reached that has no better adjacent vertices, because the convexity of the feasible region then implies that this vertex is optimal.

The simplex method is an exponential-time algorithm (in the worst case). However, it consistently has proven to be very efficient in practice. Running time tends to grow approximately with the cube of the number of functional constraints, and less than linearly with the number of variables. Problems with many thousands of functional constraints and a larger number of decision variables are routinely solved. One key to its efficiency on such large problems is that the path followed generally passes through only a tiny fraction of all vertices before reaching an optimal solution. The number of iterations (vertices traversed) generally is of the same order of magnitude as the number of functional constraints.

The running time of the simplex method also is greatly affected by the degree of sparsity of the matrix of constraint coefficients, where the measure of sparsity is the proportion of the coefficients that are not zero. Having a very sparse coefficient matrix (say, less than 1%) can greatly accelerate the simplex method.

There also exist useful variants of the simplex method, including especially the dual simplex method, that sometimes are used to solve linear-programming problems. (Using the terminology introduced at the beginning of the next section, the dual simplex method operates on the primal problem as if the simplex method is being applied simultaneously to the dual problem).

In addition, specialized versions of the simplex method also are available for exploiting the special structure in some of the special types of linear-programming problems described in the preceding section. In particular, the network-simplex method does this for the minimum-cost network-flow problem and the transportation-simplex method does it for the transportation problem. A variety of special algorithms also are available for the assignment problem, the maximum-flow problem, and the shortest-path problem. Therefore, even though the general simplex method can solve huge instances of these problems, these special purpose algorithms can solve even vastly larger instances.

Any of the various textbooks on linear programming cited in the references will provide additional details about the simplex method and these related algorithms.

Some 37 years after the invention of the simplex method, N. Karmarkar (1984) created great excitement in the operations research/management science community by announcing a new polynomial-time algorithm for linear programming, along with claims of being many times faster than the simplex method. Actually, the first polynomial-time algorithm for linear programming had been announced earlier by L. G. Khachiyan (1979), but his ellipsoid method proved to be not nearly competitive with the simplex method in practice. Karmarkar's algorithm moves through the interior of the feasible region until it converges to an optimal solution, and so is referred to as an

interior-point method. The announcement did not include details needed for computer implementation.

Following Karmarkar's announcement, there was a long flurry of research activity to fully develop and refine similar interior-point methods, along with sophisticated computer implementations. The application of these methods to linear programming now has reached a high level of sophistication. These methods commonly are called barrier methods or barrier algorithms because they are based on introducing a logarithmic barrier function. A specific barrier algorithm then may be given a specific name to identify its main features. For example, the primal-dual predictor-corrector algorithm developed by Mehrotra (1992) established a structure that has commonly been adopted by subsequent algorithms. Ye (1997), Vanderbei (2008), and Luenberger and Ye (2008) provide further details about the interior-point approach.

A key feature of the interior-point approach is that both the number of iterations (trial solutions) and total running time tend to grow very slowly (even more slowly than for the simplex method) as the problem size is increased. Therefore, the best implementations of this approach tend to become faster than the simplex method (or the dual simplex method) for relatively large problems. This is not always true, because the efficiency of each approach depends greatly in different ways on the special structure in each individual problem. Indeed, one of the by-products of the emergence of the interior-point approach has been a major renewal of efforts to improve the efficiency of computer implementations of the simplex method and its variants. Impressive progress has been made. Consequently, when tests have been conducted to determine when a leading barrier algorithm, the simplex method, or the dual simplex method will solve various huge problems more quickly, the dual simplex method or simplex method occasionally wins. As time goes on, improving computer technology (such as massive parallel processing) will substantially increase the size of problems that any of the algorithms can solve.

A considerable number of excellent software packages for linear programming and its extensions now are available to fill a variety of needs. Leading packages include CPLEX, Express-MP, Gurobi, and LINDO. Frontline Systems also has excellent solvers, including its Risk Solver Platform, for use with Excel spreadsheets.

As mentioned earlier, when dealing with large linear-programming problems, modeling languages also are needed to efficiently input, formulate, and manage the model. The available modeling languages include AMPL, MPL, OPL, GAMS, and LINGO. These languages are designed to be integrated with the kinds of solvers mentioned in the preceding paragraph.

Duality Theory and Postoptimality Analysis

Associated with any linear-programming problem is another linear-programming problem called the dual. Furthermore, the relationship between the original problem (called the primal) and its dual is a symmetric one, so that the dual of the dual is the primal. For example, consider the two related linear-programming models shown below in matrix notation (where $\boldsymbol{A}$ is a matrix, $\boldsymbol{c}$ and $\boldsymbol{y}$ are row vectors, $\boldsymbol{b}$, $\boldsymbol{x}$, and the null vector $\boldsymbol{0}$ are column vectors, all with compatible dimensions, and $\boldsymbol{x}$ and $\boldsymbol{y}$ are the decision vectors):

Maximize $\boldsymbol{cx}$	Minimize $\boldsymbol{yb}$
subject to: $\boldsymbol{Ax} \leq \boldsymbol{b}$	subject to: $\boldsymbol{yA} \geq \boldsymbol{c}$
and $\boldsymbol{x} \geq \boldsymbol{0}$.	and $\boldsymbol{y} \geq \boldsymbol{0}$.

For each of these problems, its dual is the other problem.

There are many useful relationships between the primal and dual problems, so the dual provides considerable information for analyzing the primal. This is especially helpful when conducting postoptimality analysis, i.e., analysis done after finding an optimal solution for the initial validated version of the model. A key part of most linear-programming studies, this analysis addresses a variety of what-if questions of interest to the decision makers. The purpose is to explore various scenarios about future conditions that may deviate from the initial model. The dual simplex method frequently is helpful for quickly re-optimizing these revised models.

Although the parameters of the given linear-programming model are treated as constants, they frequently represent just best estimates of a quantity whose true value may turn out to be quite

different. A key part of postoptimality analysis is sensitivity analysis, an investigation of the parameters to determine which ones are sensitive parameters, i.e., those that change the optimal solution if a small change is made in the given parameter value, and exploring the implications. For certain parameters, the decision makers may have some control over its value (e.g., the amount of a resource to be made available), in which case sensitivity analysis guides the decision on which value to choose. An extension of sensitivity analysis called parametric programming enables systematic investigation of simultaneous changes in various parameters over ranges of values.

Fletcher et al. (1999) present an interesting case study of how an OR team at the Pacific Lumber Company made extensive use of detailed sensitivity analysis to develop a sustained yield plan for the company's entire landholding. This plan is credited with increasing the company's present net worth by over $398 million while also generating a better mix of wildlife habitat acres.

Extensions of the simplex method are well suited for performing these kinds of postoptimality analysis. However, this is less true for interior-point methods. Therefore, even when an interior-point method is used to find an optimal solution, a switch may be made to the simplex method for subsequent analysis.

When there is substantial uncertainty about what the true values of the parameters will turn out to be, it may be necessary to use a different analysis approach, called linear programming under uncertainty, in which some or all the parameters are treated as random variables. This is especially pertinent when planning must be done for multiple time periods into an uncertain future. For example, Infanger (1993) discusses solving large-scale multi-stage stochastic linear programs.

Further Reading

Dantzig (1982) describes some of the early history of linear programming. Gass (1990) gives an entertaining introduction to the field. Hillier and Lieberman (2010) expand on all the topics mentioned here at an elementary level, and F.S. Hillier and M.S Hillier (2011) emphasize the application of linear programming from a managerial viewpoint. Dantzig (1963) provides the classic textbook on the theory of linear programming. Other excellent textbooks on linear programming and its extensions include Bertsimas and Tsitsiklis (1997), Dantzig and Thapa (1997, 2003), Vanderbei (2008), Luenberger and Ye (2008), Murty (2010), and Bazaraa, Jarvis and Sherali (2010), Marsten, Subramanian, Saltzman, Lustig, and Shanno (1990) discuss the basic concepts underlying interior-point methods.

See

- Algebraic Modeling Languages for Optimization
- Assignment Problem
- Basis
- Computational Complexity
- Density
- Duality Theorem
- Game Theory
- Hierarchical Production Planning
- Integer and Combinatorial Optimization
- Interior-Point Methods for Conic-Linear Optimization
- Mathematical Model
- Model Management
- Multiplier Vector
- Nonlinear Programming
- Parametric Programming
- Postoptimal Analysis
- Primal Problem
- Sensitivity Analysis
- Simplex Method (Algorithm)
- Simplex Tableau
- Stochastic Programming
- Transportation Problem
- Verification, Validation, and Testing of Models

References

Bazarra, M. S., Jarvis, J. J., & Sharali, H. D. (2010). *Linear programming and network flows* (4th ed.). New York: Wiley.

Bertsimas, D. M., & Tsitsiklis, J. N. (1997). *Linear optimization*. Belmont, MA: Athena Scientific.

Bixby, A., Downs, B., & Self, M. (2006). A scheduling and capable-to-promise application for swift & company. *Interfaces, 36*(1), 39–50.

Camm, J. D., Chorman, T. E., Dill, F. A., Evans, J. R., Sweeney, D. J., & Wegryn, G. W. (1997). Blending OR/MS, judgment, and GIS: Restructuring P&G's supply chain. *Interfaces, 27*(1), 128–142.
Dantzig, G. B. (1963). *Linear programming and extensions*. Princeton, NJ: Princeton University Press.
Dantzig, G. B. (1982). Reminiscences about the origins of linear programming. *Operation Research Letters, 1*, 43–48.
Dantzig, G. B., & Thapa, M. N. (1997). *Linear programming 1: Introduction*. New York: Springer.
Dantzig, G. B., & Thapa, M. N. (2003). *Linear programming 2: Theory and extensions*. New York: Springer.
Fletcher, L. R., Alden, H., Holmen, S. P., Angelis, D. P., & Etzenhouser, M. J. (1999). Long-Term forest ecosystem planning at pacific lumber. *Interfaces, 29*(1), 90–112.
Gass, S. I. (1990). *An illustrated guide to linear programming*. New York: Dover.
Hillier, F. S., & Hillier, M. S. (2011). *Introduction to management science: A modeling and case studies approach with spreadsheets* (4th ed.). Burr Ridge, IL: Irwin/McGraw-Hill.
Hillier, F. S., & Lieberman, G. J. (2010). *Introduction to operations research* (9th ed.). New York: McGraw-Hill.
Holloran, T. J., & Byrne, J. E. (1986). United airlines station manpower planning system. *Interfaces, 16*(1), 39–50.
Infanger, G. (1993). *Planning under uncertainty: Solving large-scale stochastic linear programs*. Danvers, MA: Boyd and Fraser.
Ireland, P., Case, R., Fallis, J., Van Dyke, C., Kuehn, J., & Meketon, M. (2004). The Canadian pacific railway transforms operations by using models to develop its operating plans. *Interfaces, 34*(1), 5–14.
Karmarker, N. K. (1984). A New Polynomial-time Algorithm for Linear Programming. *Combinatorica, 4*, 373–395.
Khachiyan, L. G. (1979). A polynomial algorithm for linear programming. *SSSR Doklady Akademii Nauk, 244*, 1093–1096. Translated in Soviet Math. Doklady 20 (1979), 191–194.
Klingman, D., Phillips, N., Steiger, D., & Young, W. (1987). The successful deployment of management science throughout Citgo Petroleum Corporation. *Interfaces, 17*(1), 4–25.
Leachman, R. C., Kang, J., & Lin, Y. (2002). SLIM: Short cycle time and low inventory in manufacturing at Samsung Electronics. *Interfaces, 32*(1), 61–77.
Lee, E. K., & Zaider, M. (2008). Operations research advances cancer therapeutics. *Interfaces, 38*(1), 5–25.
Luenberger, D. G., & Ye, Y. (2008). *Linear and nonlinear programming* (3rd ed.). New York: Springer.
Marsten, R., Subramanian, R., Saltzman, M., Lustig, I., & Shanno, D. (1990). Interior point methods for linear programming: Just call Newton, Lagrange, and Fiacco and McCormick! *Interfaces, 20*(4), 105–116.
Mehrotra, S. (1992). On the implementation of a primal-dual interior point method. *SIAM Journal on Optimization, 2*(4), 575–601.
Murty, K. G. (2010). *Optimization for decision making: Linear and quadratic models*. New York: Springer.
Vanderbei, R. J. (2008). *Linear programming: Foundations and extensions* (3rd ed.). New York: Springer.
Ye, Y. (1997). *Interior point algorithms*. New York: Wiley.

Linear-Fractional Programming Problem

The linear-fractional programming problem is one in which the objective to be maximized is of the form $f(\boldsymbol{x}) = (\boldsymbol{cx} + \alpha)/(\boldsymbol{dx} + \beta)$ subject to $\boldsymbol{Ax} \leq \boldsymbol{b}$, $\boldsymbol{x} \geq \boldsymbol{0}$, where α and β are scalars, $\boldsymbol{c}$ and $\boldsymbol{d}$ are row vectors of given numbers, and $\boldsymbol{b}$ is the right-hand-side vector. The problem can be converted to an equivalent linear programming problem by the translation $\boldsymbol{y} = \boldsymbol{x}/(\boldsymbol{dx} + \beta)$, provided that $\boldsymbol{dx} + \beta$ does not change sign in the feasible region.

See

► Fractional Programming

Lipschitz Continuous

A function $f(x)$ is said to be Lipschitz continuous if there exists a real constant $K > 0$ (called the Lipschitz constant) such that for every pair of points $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, $||f(\boldsymbol{x}_1) - f(\boldsymbol{x}_2)|| \leq K||\boldsymbol{x}_1 - \boldsymbol{x}_2||$. If $K < 1$, then the function is called a contraction.

Little's Law

Susan Albin
Rutgers, The State University of New Jersey, Piscataway, NJ, USA

Little's Law, among the most fundamental and useful formulas in queueing theory, relates the number of customers in a queueing system to the waiting time of customers for a system in steady state as

$$L = \lambda W$$

- $L =$ The average number of customers in the system including customers in service
- $\lambda =$ The average arrival rate of customers to the system; and
- $W =$ The average time a customer spends in the system including the time in service

An alternate form of Little's Law addresses only the customers in the waiting line, or queue, i.e.,

$$L_q = \lambda W_q$$

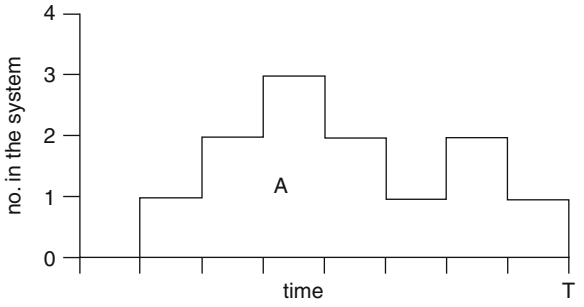
- L_q = The average number of customers in the queueing (excluding customers in service);
- λ = The average arrival rate of customers to the queueing system; and
- W_q = The average time that a customer spends in the queueing (excluding the time in service).

Little's Law, formally proven in Little (1961) and simplified in Stidham (1974), is remarkably general, requiring only that the queueing is ergodic and that no service needs are artificially created or destroyed (i.e., the system is work conserving). The result holds for any arrival process, service-time distribution, and number of servers. It applies for all queueing disciplines, with the customers not necessarily served in order of arrival, and for a specific class of customers that are distinguished from others by priority or some other characteristic. Little's formula holds for every infinite sample path realization of the queueing system, and it is approximately valid in finite intervals, with the accuracy increasing as the interval increases.

In the study of queueing, whether by mathematical analysis, simulation or direct data collection, it is often simpler to find either the average number in system or the average waiting time. Once the simpler one has been found, Little's Law gives the other. For example, in an operating manufacturing system, if average time in the system (lead time) is simpler to estimate from data, Little's Law can be used to estimate the average number of parts in the system (in process inventory).

An outline of a proof of Little's Law is based on depicting a sample path of the number in the system over an interval of time T for a steady-state queueing system with arrival rate λ (Fig. 1). The number of customer-minutes spent in the system equals A, the area under the curve. The average number of customers that arrive in the interval is λT (approximately); thus the average number of minutes in the system per customer is $W = A/(\lambda T)$. The average number of customers in the system $L = A/T$. Manipulating the two equations, taking limits, and accounting for end effects yields Little's Law.

An outline of a proof of Little's Law is based on depicting a sample path of the number in the system over an interval of time T for a steady-state queueing system with arrival rate λ (see Fig. 1). The number of customer-minutes spent in the system equals A, the area under the curve. The average number of customers that arrive in the interval is λT (approximately); thus the average number of minutes in the system per customer is $W = A/(\lambda T)$. The average number of customers in the system $L = A/T$. Manipulating the two equations, taking limits, and accounting for end effects yields Little's Law.



Little's Law, Fig. 1 Sample path realization of the number in the system over time

See

▶ Queueing Theory

References

Little, J. D. C. (1961). A proof for the queueing formula: $L = \lambda W$. *Operations Research, 9*, 383–387.

Little, J. D. C. (2011). Little's Law as viewed on its 50th anniversary. *Operations Research, 59*, 536–529.

Stidham, S., Jr. (1974). A last word on $L = \lambda W$. *Operations Research, 22*, 417–421.

Little's Law in Distributional Form

L. D. Servi
The MITRE Corporation, Bedford, MA, USA

Since Little's Law first appeared in 1961, its simplicity and importance have established it as a basic tool of queueing theory. Little's Law relates the average number of customers in a system, N, with the average

time in the system, T, under very broad conditions. For example, Keilson and Servi (1988) have demonstrated that for many systems, the relationship between the queueing length and the time in the system can be characterized beyond just their average value.

This is possible, however, if a class of customers arrives according to a Poisson process, is served first-in, first-out (FIFO) within the class, and is processed as either

1. An ordinary single-server queueing,
2. A single-server queueing with one or more classes of priority which processes each class according to a preemptive-resume, preemptive-repeat, or nonpreemptive discipline,
3. A vacation model system, where the server takes one or more vacations when the queueing is depleted,
4. A polling system, where a single server moves cyclically between (real or virtual) queueing, either serving the customers at the queueing to exhaustion, employing a Bernoulli schedule, or serving at most K customers at a queueing before moving on, or
5. An $M/G/G/\ldots G/1$ tandem queueing system, where the output of one queueing is the input of another and the service times at successive queueing are i.i.d. service times for successive arrivals.

More precisely, Keilson and Servi (1988) demonstrated that, if for a given class of customers,

(C-1) The arrival process is Poisson with rate λ,

(C-2) All arriving customers enter the system and remain in the system until served,

(C-3) The customers leave the system one at a time in order of arrival, and

(C-4) For any time t, the arrival process after time t, and the time in the system of any customer arriving before time t, are statistically independent,

then the relationship between the probability distribution of the number in the system and the time in the system follows the simple formula

$$\pi_N(u) = \alpha_T(\lambda - \lambda u) \tag{1}$$

where $\pi_N(u) = E[u^N]$ is the probability generating function of N and $\alpha_T(s) = E[e^{sT}]$ is the Laplace transform of the density of T.

Since $d^n\pi_N(u)/du^n = E[N(N-1)\ldots(N-n+1)]$ for $u = 1$ and $d^n\alpha_T(s)/ds^n = (-1)^n E[T^n]$ for $s = 0$, one can relate the moments of queueing lengths to the moments of the time in the system by computing successive derivatives of (1) with respect to u and then evaluating at $u = 1$. For example,

$$\begin{aligned}
E[N] &= E[\lambda T] \\
E\left[N^2\right] &= E\left[(\lambda T)^2\right] + E[\lambda T] \\
E\left[N^3\right] &= E[\lambda T] + 3E\left[(\lambda T)^2\right] + E\left[(\lambda T)^3\right] \\
E\left[N^4\right] &= E[\lambda T] + 7E\left[(\lambda T)^2\right] + 6E\left[(\lambda T)^3\right] + E\left[(\lambda T)^4\right] \\
E\left[N^5\right] &= E[\lambda T] + 15E\left[(\lambda T)^2\right] + 25E\left[(\lambda T)^3\right] + 10E\left[(\lambda T)^4\right]
\end{aligned} \tag{2}$$

The first of these equations is the familiar Little's Law. As is the case of the Pascal Triangle, there is a simple relation between the coefficients. Specifically, one can show that

$$E[N^n] = \sum_{m=1}^{n} S(n,m) E[\lambda T]^m \tag{3}$$

where $S(u, m)$ is a Stirling number of the second kind defined by the recursion $S(n+1,m) = mS(n,m) + S(n,m-1)$ for $n+1 \geq m \geq 1$, $S(n,0) = S(n,n+1) = 0$ for $n \geq 1$ and $S(1,1) = 1$ (Abramowitz and Stegun 1972).

Similarly,

$$E[(\lambda T)^n] = \sum_{m=1}^{n} \bar{S}(n,m) E[N^m]$$

where $\bar{S}(n,m)$ are Stirling numbers of the first kind which satisfy $\bar{S}(n,m-1) = \bar{S}(n,m-1) - n\bar{S}(n,m)$ for $n+1 \geq m \geq 1$, $\bar{S}(n,0) = \bar{S}(n,n+1) = 0$ for $n \geq 1$ and $\bar{S}(1,1) = 1$.

The first two equations of (2) imply the simple but non-intuitive formula

$$\frac{Var[N]}{E[N]} = \frac{Var[\lambda T]}{E[\lambda T]} + 1.$$

The system could refer to the queueing and the pool of customers in service or exclusively to the queueing. In the latter case, additional systems satisfy conditions (C-1)–(C-4). For example, for a multi-server system,

the customers do not leave the system consisting of the queueing and the pool of customers in service on a first-in, first-out basis [and hence violate condition (C-3)]. However, if the system refers exclusively to the queueing, then condition (C-3) is satisfied.

These results have been generalized, for example, to systems with non-Poisson arrivals (Bertsimas and Mourtzinou 1997), to systems operating under heavy traffic (Szczotka 1992), to systems having batch arrivals (Takahashi and Miyazawa 1994), and has been used as the basis to derive explicit formulae for the distribution of the number in the system (or queueing) as well as the time in the system (or queueing) for a number of more classical systems (Keilson and Servi 1990).

See

- ▶ Little's Law
- ▶ Queueing Theory

References

Abramowitz, M. & Stegun, I. A. (1972). *Handbook of mathematical functions with formulas, graphs, and mathematical tables*. National Bureau of Standards, U.S. Government Printing Office. 824–825.

Bertsimas, D., & Mourtzinou, G. (1997). Transient laws of non-stationary queueing systems and their applications. *Queueing Systems, 25*, 115–155.

Keilson, J., & Servi, L. D. (1988). A distributional form of Little's Law. *Operation Research Letters, 7*, 223–227.

Keilson, J., & Servi, L. D. (1990). The distributional form of Little's Law and the Fuhrmann-Cooper decomposition. *Operations Research Letters, 9*, 239–247.

Little, J. (1961). A proof of the theorem $L = \lambda W$. *Operations Research, 8*, 383–387.

Szczotka, W. (1992). A distributional form of Little's Law in heavy traffic. *Annals Probability, 20*, 790–800.

Takahashi, Y., & Miyazawa, M. (1994). Relationship between queueing-length and waiting time distributions in a priority queueing with batch arrivals. *Journal of the Operations Research Society of Japan, 37*, 48–63.

Local Balance Equations

- ▶ Detailed Balance Equations
- ▶ Queueing Theory

Local Improvement Heuristic

A heuristic rule which examines all the solutions that are closely related to a given initial solution and is guaranteed to reach at least a local optimum.

See

- ▶ Heuristic Procedure
- ▶ Local Optimum

Local Maximum

A function $f(x)$ defined over a set of points S is said to have a local maximum at a point $\boldsymbol{x}_0$ in S if $f(\boldsymbol{x}_0) \geq f(\boldsymbol{x})$ for all $\boldsymbol{x}$ in a neighborhood of $\boldsymbol{x}_0$ in S. The point $\boldsymbol{x}_0$ is referred to as a local optimum (maximum).

See

- ▶ Global Maximum (Minimum)
- ▶ Nonlinear Programming
- ▶ Quadratic Programming

Local Minimum

A function $f(\boldsymbol{x})$ defined over a set of points S is said to have a local minimum at a point $\boldsymbol{x}_0$ in S if $f(\boldsymbol{x}_0) \leq f(\boldsymbol{x})$ for all $\boldsymbol{x}$ in a neighborhood of $\boldsymbol{x}_0$ in S. The point $\boldsymbol{x}_0$ is referred to as a local optimum (minimum).

See

- ▶ Global Maximum (Minimum)
- ▶ Nonlinear Programming
- ▶ Quadratic Programming

Local Optimum

▶ Local Maximum
▶ Local Minimum

Local Solution

A best solution in a feasible neighborhood.

Location Analysis

Charles ReVelle[1] and Vladimir Marianov[2]
[1]The Johns Hopkins University, Baltimore, MD, USA
[2]Pontificia Universidad Católica de Chile, Santiago, Chile

Introduction

The term location analysis refers to the development of formulations and algorithms/methodologies to site facilities of diverse kinds in a spatial or geographic environment. The facilities may be sited with relation to demand points, supply points, or with respect to one another. Although facility layout falls within this definition, this topic is not generally considered under the rubric of location analysis. Common descriptive terms for location analysis are deployment, positioning, and siting, although these terms are actually the outcome that follows the execution of a formulation or algorithm.

Location settings may be classified into two broad categories: planar problems and network problems. Planar problems typically assume that the distances between facilities and demand points, supply points or other facilities are given by a metric, a formula that calculates distance between points based on their coordinates in space. Network problems, in contrast, assume that travel can only occur on an underlying network and that distances are the lengths (or cost) of the shortest paths between the particular points on the network. A further distinction between these categories is provided by the assumption in most planar problems of an infinite solution space, that is, that facilities can be sited anywhere on the plane, perhaps subject to exclusion areas or regions. These planar problems are most often non-linear optimization problems and more abstract in their application than network-based problems. In contrast to the infinite solution space assumed by most planar problems, all but a few network problems restrict facilities to sites that have been specified in advance as eligible to house those facilities. The network problems tend to be linear zero-one optimization problems and so pose challenges in their resolution to integers. First, planar problems and approaches to them will be discussed; followed by a discussion about network location formulations and their solution.

Planar Location Problems

The most famous of the planar problems and the first location problem to be posed historically is the minimum Euclidean single facility location problem first stated by Fermat as a mathematical problem: "Given three points in the plane, find a fourth such that the sum of its distances to the three given points is a minimum" (Kuhn 1967). It is often referred to as the Weber problem, after the German economist who first discussed it in economic terms (Weber 1909). The minimum problem considers points dispersed on the plane that send items to or receive finished product from some central factory or facility. The problem seeks the central point that minimizes the sum of weights (quantities) times the distances to all dispersed points. The problem assumes that the Euclidean distances separate the dispersed points and the central point, that the central point can be anywhere on the plane, and that a weight or loading is associated with each of the dispersed points. An iterative solution method that can be shown to converge to an optimal solution was offered in the 1930s, lost to view, and rediscovered in the early 1960s by several independent investigators. In the minisum multiple facility problem (the multi-Weber problem), a number of central facilities are to be sited, each one associated with a cluster or partition of the dispersed points.

An allocation problem arises, i.e., the problem of deciding which facility serves each dispersed point.

The history of the minisum problem is reviewed in Wesolowsky (1993). Only in the early 1990s has this problem yielded to exact methods, followed by heuristics and metaheuristics (Brimberg et al. 2008).

While the Weber problem in its single and multi-facility forms utilizes the Euclidean metric for distances, the minisum rectilinear problem utilizes the Manhattan or rectilinear metric for distances and minimizes the sum of weights times these distances to the central point. The rectilinear distance between two points is the sum of the horizontal and vertical separation of the points. Because the problem can be reduced to the choice among a set of eligible points, the multi-facility rectilinear minisum problem yields either to heuristics or to the linear integer-programming formulation used for the *p*-median problem, a problem that will be discussed under network location models. When the classic metrics are set aside, solution of the minisum problem generally becomes more difficult, except in the case of minimizing the weighted sum of squared distances, in which case the single facility minisum solution is simply the centroid.

A second important objective setting in planar problems is the siting of a single facility under the objective of minimizing the maximum distance that separates any demand/supply point from the central facility. The problem may utilize either of the two classic metrics, Euclidean or rectilinear. No matter the number of dispersed points, the minimax single facility location problem with rectilinear distances yields to either a geometric solution or to a four-constraint linear program. The minimax single facility location problem with Euclidean distances is a nonlinear-programming problem, but can also be solved by a geometric argument. Multi-facility versions of the planar minimax location problems may yield to heuristics resembling those applied to the *p*-median problem. A good general reference dealing in part with planar location problems is the text of Love et al. (1988), Plastria (1995) provides a comprehensive review for the researcher in planar location.

It is worth mentioning that researchers in continuous location, seeking a greater realism in their problems, have sought to project the most likely real distances on a road network between a pair of points given the spatial coordinates of these points. This literature is reviewed in Brimberg and Love (1995).

Network Location Problems

In contrast to the use of formula-based metrics for the siting of facilities on a plane, network location problems always measure distances across the links of the network. Interestingly, the assumption of an infinite solution space can be made in network-based location problems as well. That is, the infinite solution space would consist of all the points on every arc of the network. For some problems, including the *p*-median, the solution space can be reduced without loss of optimality from all the points on all the arcs to a limited number of eligible points when the triangle inequality holds throughout the network. Many network problems simply assume a prespecified set of eligible facility sites based on needed characteristics of such points, such as transportation infrastructure, availability of lots or warehouse space, etc.

Within network location research two distinct foci are found. The first is cost minimizing/profit maximizing siting that is goods-oriented, an activity especially of the manufacturing and distribution industries. The second is people or service-oriented siting, an activity mostly of government at a number of levels from local to national, but also of private companies. The divisions are not perfect, as it will be seen, but are, at the least, useful for discussion purposes. These two settings will be taken up in that order followed by presentations of some variations and adaptations of these classes.

Goods-Oriented Siting

By far, the problem setting considered most extensively in the goods oriented location category is the simple plant location problem (SPLP). The problem assumes that an unknown number of plants are to be sited to manufacture product for distribution to a number of spatially dispersed demand points. The plants have no limit as to the amount manufactured, and each point must be fully supplied with its demand. The objective is the

minimization of the total of manufacturing cost and distribution cost. Manufacturing includes a fixed opening cost and an expansion cost that can be linear or nonlinear. The problem may be stated mathematically as:

$$\text{minimize} \quad z = \sum_{i=1}^{m}\sum_{i=1}^{n} c_{ij}x_{ij} + \sum_{i=1}^{m} f_i y_i$$

subject to :

$$\sum_{i=1}^{m} x_{ij} = 1, \qquad j = 1, \ldots, n,$$

$$y_i - x_{ij} \geq 0, \qquad i = 1, \ldots, m; j = 1, \ldots, n,$$

$$x_{ij}, y_i \in \{0, 1\}, \quad i = 1, \ldots, m; j = 1, \ldots, n.$$

i = the index of eligible plant sites of which there are m;
j = index of demand points of which there are n;
f_i = opening cost for a plant at i;
c_{ij} = cost to deliver j's full demand from i, including the production cost at i;
$y_i \in \{0, 1\}$, it is 1 if a plant opens at i and 0 otherwise; and
$x_{ij} \in \{0, 1\}$, it is 1 if i delivers j's full demand and 0 otherwise.

The above problem formulation is due to Balinski (1965), and is one of several formulations possible for the SPLP. It is presented here because it is the basis for a number of solution methods.

The SPLP has attracted attention since the 1950s when heuristics were first suggested. In the 1960s, Balinski offered his formulation of the problem but dismissed it as unreliable. In addition, several branch and bound algorithms were created to solve the SPLP, but these algorithms proved impractical for large problems. In the mid-1970s, Bilde and Krarup (1977) and Erlenkotter (1978) both proposed dual ascent algorithms for the SPLP; the basic algorithm proposed by these two sets of investigators has proved to be capable of handling relatively large problems. Morris (1978) investigated 500 randomly generated plant location problems and found that if the formulation above were solved as a linear program (without integer requirements on any of the variables) that 96% of the problems so solved presented with all zero-one variables. Morris' experience thus suggested that linear programming alone was a powerful technique for the SPLP formulation that Balinski had abandoned. The problem has since been successfully pursued by Lagrangian relaxation by Galvão (1989) and Korkel (1989), who modified the dual ascent algorithm referred to above to solve remarkably large problems.

While the SPLP has attracted considerable attention, a related form, the capacitated plant location problem (CPLP), languished until the late 1980s. The CPLP sets limits on the amount that could be manufactured at any site, but in all other respects is the same as the SPLP. First attacked by Davis and Ray (1969), the problem later received attention from Pirkul (1987), who provided both references to prior work and a solution algorithm based on Lagrangian relaxation. The CPLP also describes a problem in solid waste management in which waste is generated at population nodes and must be disposed of at sanitary landfills with limited capacities. Landfills are to be sited in this problem statement.

Many other plant location style problems can be stated. A maximum profit version of the SPLP is one such statement. The time dimension has been incorporated in a number of models, Melo et al. (2005). Multiple products can be treated as well. Another line of research focuses on the representation of the cost, since in many cases there are economies of scale or costs that are piecewise linear. Inventory, as well as other logistics costs can be also integrated in these models, see Snyder et al. (2007). Finally, demands, prices, and costs can be viewed as random, leading to stochastic versions of the plant location problem. The SPLP has been not only used for goods-oriented siting, but also for the design of telecommunications networks; in particular, for solving a problem called the Concentrator Location Problem, whose mathematical structure is identical to that of the warehouse or plant location problem. Shen (2007) surveys integrated supply chain design models.

Public Service-Oriented Siting

Nearly all of the plant location problems – excluding the concentrator location problem – emphasize the flow/movement of goods. In contrast, service oriented siting problems focus on the accessibility of people to services or services to people. Flow/movement is part of the equation in some of the models, but simple geographic coverage can suffice in others.

The same two objectives treated under planar problems, minisum and minimax, have also been considered for network location problems of service

siting. The minisum network location problem is known as the *p*-median problem; the minimax network location problem is known as the *p*-center problem. Both were posed together in seminal papers by Hakimi (1964, 1965). He also proved that there is always an optimal solution considering location only at nodes of the network.

The *p*-median problem, which seeks the minimum cost assignment of each population node to one of *p* facilities, resembles the SPLP in all but one modeling aspect. Indeed, so strong is the resemblance of *p*-median to the simple plant location model that the same algorithms may be used for solution of both with minor adaptation, Galvão (1989). The single difference between the two models is easy to explain once a mathematical-programming formulation of the *p*-median is offered. The *p*-median problem seeks to site *p* facilities in such a way that the least total of people times distance traveled to the assigned facility is achieved. Division of this objective by the total of population reveals that minimization of the total population-miles objective also minimizes the average distance that people travel to service. Travel/assignment is always assumed to the closest among the *p* facilities.

The *p*-median problem may be formulated as:

$$\text{minimize} \quad Z = \sum_{i=1}^{n} \sum_{j=1}^{n} a_i d_{ij} x_{ij}$$

subject to :

$$\sum_{j=N_i}^{n} x_{ij} = 1, \quad i = 1, 2, \ldots, n,$$

$$x_{jj} - x_{ij} \geq 0, \quad i, j = 1, 2, \ldots, n; i \neq j,$$

$$\sum_{j=1}^{n} x_{jj} = p$$

$$x_{ij} \in \{0, 1\}, \quad i, j = 1, 2, \ldots, n,$$

a_i = relevant population at demand node i;
d_{ij} = shortest distance from node i to node j;
N = number of nodes;
P = number of facilities; and
$x_{ij} \in \{0, 1\}$; it is 1 if node i assigns to a facility at j and 0 otherwise.

It can be seen from a comparison of the *p*-median formulation and that of the SPLP that the objectives differ only in the presence or absence of fixed opening costs and their opening variables, and that the constraints differ only in the presence or absence of a constraint on the number of facilities. In all other respects, the formulations look virtually identical. If the constraint on the number of facilities in the *p*-median formulation is brought to the objective with a multiplier λ, the objective becomes

$$\sum_{i=1}^{n} \sum_{j=1}^{n} a_i d_{ij} x_{ij} + \sum_{j=1}^{n} \lambda x_{jj}.$$

The subscripts reflect flow between central facilities and demand points. The *p*-median is now fully equivalent to an SPLP with equal opening costs, thus making all the techniques for solution of the SPLP available for solution of the *p*-median. Ranging the multiplier λ in the *p*-median is equivalent to trading off people miles against the number of facilities by use of the weighting method of multi-objective programming. Among the methods available for the SPLP that can be used for the *p*-median are relaxed linear programming (ReVelle and Swain 1970), the dual ascent methodology (Bilde and Krarup 1977; Erlenkotter 1978) and Lagrangian relaxation (Galvão 1989). A number of other researchers have used heuristics for the *p*-median problem; a listing of many of the early methods for the *p*-median problem appeared in ReVelle et al. (1977). Newer and more effective heuristic and metaheuristic methods are reviewed in Mladenovic et al. (2007) and Reese (2006). As the SPLP, the *p*-median also has a capacitated version in which each facility can serve up to a certain number of people.

While the *p*-median problem attracted considerable attention, researchers found its focus on the average condition of population accessibility to be limiting. Concern for those worst off relative to their distance to the nearest facility, that is, for the maximum distance or time separating population centers from service, gave rise to another concept, that of coverage. A population node is considered to be covered, i.e., adequately served, if it has a facility sited within some maximum distance or time; that is, sited within a time standard. Coverage can either be required for all demand points within the standard, or maximization of demand covered can be sought, giving rise to a host of new problems, the earliest of which is the location set covering problem (LSCP).

The LSCP seeks to position the least number of facilities so that every point of demand has at least

one facility sited within the time or distance standard. The problem can be stated as a linear zero-one programming problem as follows:

$$\text{minimize} \quad z = \sum_{j \in J} x_j$$

$$\text{subject to :} \sum_{j \in N_i} x_j \geq 1 \quad \forall i \in I,$$

$$x_j \in \{0, 1\} \quad \forall j,$$

i, I = index and set of demands;
j, J = index and set of eligible sites for facilities;
$x_j \in \{1, 0\}$, 1 if a facility placed at j and 0 otherwise;
d_{ji} = the shortest distance (or time) from site j to demand point i;
S = the maximum distance (or time) that a demand point can be from its nearest facility; and
$N_i = \{j | d_{ji} \leq S\}$ = the set of facility sites eligible to serve demand point i, by virtue of being within S of i.

While general set covering problems may require integer-programming algorithms to solve them, the LSCP appears to possess special properties. In particular, solution of the linear-programming formulation on data from a geographic problem without any zero-one requirements produces all zero-one answers with remarkable regularity (over 95% of the time). If a set of eligible facility sites is specified in advance, the LSCP can be used to derive solutions to the p-center problem as well. The p-center problem seeks to position p facilities in such a way that the maximum distance that separates any population node from its nearest facility is as small as possible. Solutions to this problem can be found by solving a sequence of LSCP problems, with decreasing distance standards. As the distance decreases, the number of facilities required to cover all demands increases. The minimum distance standard that makes total coverage feasible with p facilities is the solution of the p-center problem (Minieka 1970). If, however, any point on any link of the network is eligible to house a facility (the infinite solution space case), the solution of the p-center problem remains open and challenging.

The LSCP, however, has several shortcomings as a meaningful problem statement. First, population is absent from the problem statement; proximity and population are not linked even though they should be. Second, all population nodes require coverage within the standard, a requirement that could and often proves very costly in terms of the number of facilities/servers required.

Recognizing these shortcomings of the LSCP, several researchers have created new models for siting that utilized the coverage concept not as a requirement but as a goal. The most widely known of these models is referred to as the maximal covering location problem (MCLP) or the partial covering problem, depending on the specific formulation. The MCLP seeks the positions for p facilities among a prespecified set of eligible points that maximize the population that has a facility sited within a distance or time standard S, that is, that maximizes the population covered. The MCLP can be stated as:

$$\text{maximize} \quad z = \sum_{i \in I} a_i y_i$$

$$\text{subject to :} \quad y_i \leq \sum_{j \in N_i} x_j \, \forall i \in I,$$

$$\sum_{j \in J} x_j = p,$$

$$x_j, y_i \in \{0, 1\}, \forall i, j,$$

where additional notation is
a_i = the population at demand node i;
$y_i \in \{1, 0\}$, it is 1 if demand i is covered by a facility within N_i and 0 otherwise; and
p = the number of facilities that can be sited.

Basically, while the LSCP is attempting to find the least resources to cover all demand nodes within the distance goal, the MCLP is attempting to distribute lesser and limited resources to achieve as much population coverage as possible (Church and ReVelle 1974).

Related Research and Extensions

The basic models described above have caught the interest of a number of researchers. The literature on the subject keeps growing.

Drezner (1987) addressed the unreliable p-median in which facilities can become inactive. Marianov and

Serra (1998) proposed models that include the effect of queuing at the facilities, while Marianov (2003) maximized the amount of people willing to get service from a facility when there is demand elasticity to travel distance and queueing. The user point of view has been embedded in the p-median by Drezner and Drezner (2007), who investigated the effect on location of considering customers' behavior, represented through gravity models.

Uncertainty has also been considered in covering models. In probabilistic covering models, the presence or availability of a vehicle or server within a time standard is not guaranteed. The probabilistic models suggest a chance constraint on vehicle availability, that is, a requirement that a vehicle be available within the time standard with a specified level of reliability, see ReVelle and Marianov (1991). The chance constraint may be a strict requirement or may be treated as a goal for each population demand node. Many of the probabilistic, as well as redundant/backup coverage models, and multiple vehicle type models were reviewed by Marianov and ReVelle (1995). A review of the applications of probabilistic coverage models to emergency systems is provided by Goldberg (2004).

A number of other lines of research within the network location setting have been pursued. Among these are hierarchical location models, models in which a hierarchy of interacting/interrelated facility types are sited. One example is the health care hierarchy in developing nations, that consists of hospitals, clinics, and remote doctors. Another is a banking system consisting of central banks, branch banks, and teller machines. Morphological relations in hierarchical systems is reviewed by Narula (1986), with a brief treatment of the topic given in Daskin (1995). Serra and ReVelle (1994) provide algorithms for the median version of these hierarchical problems where coherence of assignments is enforced. Church and Eaton (1987) present an interesting set of hierarchical models with referral between levels.

The concept of coverage has been challenged, since in some situations it does not seem reasonable to consider a demand as covered if it is within, say, 500 m from a facility, but not covered if it is at 500.1 m. Models using a gradual coverage have been reviewed by Eiselt and Marianov (2009a). In these models, the coverage function, originally a step function, can take different shapes, representing quality of coverage as a function of the distance.

Another significant line of siting research is embodied in the competitive location models in which facilities are sited in a competitive market environment with goals of capturing market share from other retailers or manufacturers, or maximizing profit in the presence of competitors. Two problems are usually solved: the follower's problem, which is to locate facilities in such a way that the market capture from existing competitors is maximized; and the leader's problem, which is to locate first in a virgin market, anticipating possible followers that will try to cannibalize the leader's market share. A review of competitive location models in continuous and discrete space is provided by Dasci (2011).

Another line of location research involves the siting of noxious facilities. Such facilities may be undesirable in of themselves and should be distant from population centers or may be required to be distant from one another. However, they usually cannot be too far, since operation costs can be prohibitive, as in garbage processing plants or jails. Several approaches have been proposed for these facilities: maximizing their distance to population; maximizing the minimum facility-population distance; compensating the population that is affected by such a facility; and expropriation. A review of obnoxious facility location problems can be found in Melanchrinoudis (2011). Another line of research addresses both location of obnoxious facilities and routing of hazardous waste (Nagy and Salhi 2007).

A problem of increasing interest is the location of hubs. As airlines and courier companies focus on logistic improvements, the location of these traffic concentration points becomes more relevant. This line of research was started by (O'Kelly 1986) and has grown towards several fronts. Hub problems can be classified into the same categories as the original location problems: hub-median, hub-location, hub-covering and hub-center problems. They can be solved on the plane (O'Kelly 1986), or on networks. Campbell et al. (2002) provide a taxonomy of hub problems. Competition and queuing effects have also been considered when locating hubs (Marianov and Serra 2003; Eiselt and Marianov 2009b).

Finally, the tools developed for location in a geographical setting can be also used in very

different spaces: to locate employees and tasks in a skill space, finding the best measurement points in the eye for glaucoma detection, and locating candidates and voters in an issue space.

Concluding Remarks

The wide variety of important applications and modeling challenges are reported in many OR/MS journals, including *Computers & Operations Research* (including *Location Science*); *European Journal of Operational Research*, *Journal of the Operational Research Society*; *IIE Transactions* and *Papers in Regional Science*. In addition, the proceedings of the triennial International Symposium on Locational Decisions (ISOLDe) have appeared in separate volumes of *Annals of Operations Research*, beginning with 1984 Boston/Martha's Vineyard conference.

See

- ▶ Facility Location
- ▶ Integer and Combinatorial Optimization
- ▶ Network
- ▶ Shortest-Route Problem
- ▶ Stochastic Programming

References

Balinski, M. (1965). Integer programming: Methods, uses and computations. *Management Science, 12*, 253–313.

Bilde, O., & Krarup, J. (1977). Sharp lower bounds and efficient algorithms for the simple plant location problem. *Annals Discrete Mathematics, 1*, 79–97.

Brimberg, J., & Love, R. (1995). Estimating distances. In Z. Drezner (Ed.), *Facility location: A survey of applications and methods* (pp. 9–31). New York: Springer.

Brimberg, J., Hansen, P., Mladenovic, N., & Salhi, S. (2008). A survey of solution methods for the continuous location-allocation problem. *International Journal of Operations Research, 5*, 1–12.

Campbell, J. F., Ernst, A. T., & Krishnamoorthy, M. (2002). Hub location problems. In Z. Drezner & H. W. Hamacher (Eds.), *Facility locations applications and theory* (pp. 373–407). New York: Springer.

Church, R., & Eaton, D. (1987). *"Hierarchical location analysis using covering objectives", in spatial analysis and location models*. New York: Van Nostrand-Rheinhold.

Church, R., & ReVelle, C. (1974). The maximal covering location problem. *Papers Regional Science Association, 32*, 101–118.

Dasci, A. (2011). Conditional location problems on networks and the plane. In H. A. Eiselt & V. Marianov (Eds.), *Foundations of location analysis* (pp. 179–206). New York: Springer.

Daskin, M. (1995). *Network and discrete location*. New York: Wiley.

Davis, P., & Ray, T. (1969). A branch-and-bound algorithm for the capacitated facilities location problem. *Naval Research Logistics Quarterly, 16*, 331–344.

Drezner, Z. (1987). Heuristic solution methods for two location problems with unreliable facilities. *Journal of the Operational Research Society, 38*, 509–514.

Drezner, Z. (Ed.). (1995). *Facility location: A survey of applications and methods*. New York: Springer.

Drezner, T., & Drezner, Z. (2007). The gravity p-median model. *European Journal of Operational Research, 179*, 1239–1251.

Eiselt, H. A., & Marianov, V. (2009a). Gradual location set covering with service quality. *Socio-Economic Planning Sciences, 43*(2), 121–130.

Eiselt, H. A., & Marianov, V. (2009b). A conditional p-hub location problem with attraction functions. *Computers and Operations Research, 36*, 3128–3135.

Erlenkotter, D. (1978). A dual-based procedure for uncapacitated facility location. *Operations Research, 26*, 992–1009.

Galvão, R. (1989). A method for solving optimality uncapacitated location problems. *Annals of Operations Research, 18*, 225–244.

Goldberg, J. B. (2004). Operations research models for the deployment of emergency services vehicles. *EMS Management Journal, 1*(1), 20–39.

Hakimi, S. L. (1964). Optimal location of switching centers and the absolute centers and medians of a graph. *Operations Research, 12*, 450–459.

Hakimi, S. L. (1965). Optimum distribution of switching centers in a communication network and some related graph theoretic problems. *Operations Research, 13*, 462–475.

Korkel, M. (1989). On the exact solution of large-scale simple plant location problems. *European Journal of Operational Research, 39*, 157–173.

Kuhn, H. (1967). On a pair of dual nonlinear programs. In J. Abadie (Ed.), *Nonlinear programming* (pp. 39–54). Amsterdam: North-Holland.

Love, R., Morris, J., & Wesolowsky, G. (1988). *Facilities location: Models and methods*. New York: North Holland.

Marianov, V. (2003). Location of multiple-server congestible facilities for maximizing expected demand, when services are non-essential. *Annals of Operations Research, 123*, 125–141.

Marianov, V., & ReVelle, C. (1995). "Siting emergency services," chapter 10 in facility location. In Z. Drezner (Ed.), *A survey of applications and methods*. New York: Springer.
Marianov, V., & Serra, D. (1998). Probabilistic maximal covering location-allocation models for congested systems. *Journal of Regional Science, 13*, 401–424.
Marianov, V., & Serra, D. (2003). Location models for airline hubs behaving as M/D/c queues. *Computers and Operations Research, 30*, 983–1003.
Melanchrinoudis, E. (2011). The location of undesirable facilities. In H. A. Eiselt & V. Marianov (Eds.), *Foundations of location analysis* (pp. 207–240). New York: Springer.
Melo, M. T., Nickel, S., & da Gama, F. (2005). Dynamic multi-commodity capacitated facility location: A mathematical modeling framework for strategic supply chain planning. *Computers and Operations Research, 33*, 181–208.
Minieka, E. (1970). The M-centre problem. *SIAM Review, 12*, 138–141.
Mladenovic, N., Brimberg, J., Hansen, P., & Moreno-Perez, J. (2007). The p-median problem: A survey of metaheuristic approaches. *European Journal of Operational Research, 179*, 927–939.
Morris, J. (1978). On the extent to which certain fixed charge depot location problems can be solved by LP. *Journal of the Operational Research Society, 29*, 71–76.
Nagy, G., & Salhi, S. (2007). Location-routing: Issues, models and methods. *European Journal of Operational Research, 177*, 649–672.
Narula, S. (1986). Minisum hierarchical location-allocation problems on a network: A survey. *Annals of Operations Research, 6*, 257–272.
Nickel, S., & Puerto, J. (2005). *Location theory: A unified approach*. New York: Springer.
O'Kelly, M. E. (1986). The location of interacting hub facilities. *Transportation Science, 20*(2), 92–106.
Pirkul, H. (1987). Efficient algorithms for the capacitated concentrates location problem. *Computers and Operations Research, 14*, 197–208.
Plastria, F. (1995). Continuous location problems. In Z. Drezner (Ed.), *Facility location: A survey of applications and methods*. New York: Springer. Chapter 11.
Reese, J. (2006). Solution methods for the p-median problem: An annotated bibliography. *Networks, 48*, 125–142.
ReVelle, C., Bigman, D., Schilling, D., Cohon, J., & Church, R. (1977). Facility location: A review of context-free and EMS models. *Health Services Research, Summer, 12*, 129–146.
ReVelle, C., & Marianov, V. (1991). A probabilistic FLEET model with individual reliability requirements. *European Journal of Operational Research, 53*, 93–105.
ReVelle, C., & Swain, R. (1970). Central facilities location. *Geographical Analysis, 2*, 30–42.
Serra, D., & ReVelle, C. (1994). Location and districting of hierarchical facilities—II heuristic solution methods. *Location Science, 2*, 63–82.
Shen, Z. J. M. (2007). Integrated supply chain design models: A survey and future research directions. *Journal of Industrial and Management Optimization, 3*, 1–27.
Snyder, L. V., Daskin, M. S., & Teo, C. P. (2007). The stochastic location model with risk pooling. *European Journal of Operational Research, 179*, 1221–1238.
Weber, A. (1909). *Uber den standort der industrien, Tübingen [Translation: (1929). Theory of the location of industries]*. Chicago: University of Chicago Press.
Wesolowsky, G. (1993). The Weber problem: History and perspectives. *Location Science, 1*, 5–23.

Logic Programming

Logic programming deals with the use of symbolic logic for problem representation and inferential reasoning. A popular logic programming language is Prolog (PROgrammation en LOGique), developed in the early 1970s by the French computer scientists, Alain Colmerauer and Philippe Roussel. Prolog has been used to develop a man-machine communication system in natural language.

See

► Artificial Intelligence

References

Bergin, T. J., Jr., & Gibson, R. G., Jr. (Eds.). (1996). *History of Programming Languages—II*. New York: ACM Press.
Cohen, J. (1988). A view of the origins and development of Prolog. *Communications of the ACM, 31*(1), 26–36.

Logical Variables

In a linear-programming problem, the set of variables that transform a set of inequalities to a set of equations are called logical variables.

See

► Linear Inequality
► Slack Variable
► Structural Variables
► Surplus Variable

Logistics and Supply Chain Management

Marius M. Solomon
Northeastern University, Boston, MA, USA

Introduction

For quite some time, logistics has accounted for a significant percentage of the U.S. gross domestic product (GDP). The Council of Supply Chain Management Professionals estimated that in 2008 the country's logistics costs were about $1.3 trillion, or 9.4% of the $13.8 trillion GDP. Year-to-year carrying costs decreased by 13.2% due to smaller inventories and lower interest rates while transportation costs rose by 2% as a result of higher fuel prices. These figures and a number of other key economic developments highlight logistics and supply chains as areas where large productivity improvements have and continue to be attained. Given the intrinsic complexity of logistics problems in today's global supply chains, such improvements could not have been achieved without the use of analytical tools, including operations research/management science (OR/MS) methodologies.

The mathematical difficulty of strategic, tactical, and operational logistics decisions and the magnitude of the potential cost savings to be achieved by utilizing OR/MS models and algorithms have attracted researchers since the early days of the field. Witness to this are the pioneering efforts of researchers in 1950s, 1960s, and 1970s. Most of the methods developed made extensive use of network models and algorithms coupled with different types of inventory techniques.

Over the last twenty five years, fueled by major developments in modeling and algorithmic methodology, constant breakthroughs in computer technology, and web-based applications, operations researchers have found logistics to be a very fertile design and implementation area. They addressed an ever increasing variety of problems with escalating complexity and size. The body of supply chain applications of OR/MS techniques also expanded at a progressively swifter pace. In what follows, the focus will be on some of the more important areas in logistics and supply chain management and, where possible, on OR/MS applications in large-scale logistics systems.

Networking and Routing

Network design and freight routing have been addressed by Braklow et al. (1992) in the context of less-than-truckload (LTL) transportation. The authors formulate the problem as a nonlinear, multicommodity network design problem. Its solution is based on a hierarchical decomposition of the overall problems into a series of optimization subproblems. The network design problem is solved using interactive optimization, where the user guides the search performed by a local improvement heuristic which adds (drops) links to (from) the load planning network. The subproblems involve the routing of the LTL shipments, of truckload shipments and of empty trailers. The former two problems are solved using shortest path algorithms, while the latter problem involves the solution of a classical linear transhipment problem. They must be reoptimized every time a change is made in the load planning network. This is performed sufficiently fast to make interactive optimization possible. The model has been used as a tactical decision tool for load planning by one of the largest LTL motor carriers. It has also been used at the strategic level to determine the location and size of new terminals.

The research of Simão et al. (2010) is illustrative of the evolution of the OR/MS methodology which had to match the increasing complexity of real-world problems due to their size and dynamism. The authors address the problem faced by a major transportation company that wanted the ability to significantly improve how it managed the dynamics of its fleet of over 6,000 long haul drivers. The issues under consideration were how to handle hiring, changes in work rules, and examine scenarios permitting the drivers to spend more time at home. Simão et al. used approximate dynamic programming (ADP) to solve this problem. ADP is a simulation-based algorithm that optimizes complex stochastic problems through iterative learning. This approach was capable to deal with both complex dynamics and multiple forms of uncertainty regarding drivers and loads and to anticipate the future impact of decisions. The model allowed the company to avoid costs and achieve savings in the millions of dollars and, at the same time, substantially improve its customer service.

While logistics encompasses a broad set of activities, two key elements are transportation

and storage. Generally, very intricate trade-offs occur between these two areas. The first focus will be on transportation issues and then address inventory matters. Transportation is in fact the most costly component of many logistics systems and supply chains. A very important segment of transportation management is the routing and scheduling of vehicles. This facet is of significant importance across land, air, and water transportation. Similar problems are also encountered in a variety of manufacturing, warehousing and service sector environments.

This area has been reviewed in several insightful surveys, including that written by Laporte (2009). The author highlights the major developments in the OR/MS methodology for the vehicle routing problem (VRP) over the last fifty years. He reviews successful exact algorithms and heuristics introduced in the literature ranging from extremely sophisticated optimal decomposition algorithms to powerful metaheuristics. His work is complemented by the books edited by Toth and Vigo (2002) and Golden et al. (2008) who put together articles spanning a multitude of VRP variants. All these sources also provide a wealth of references to research conducted over the years in the ever increasing universe of VRP problems.

While Laporte highlights the outstanding progress made by optimal algorithms, he also notes that such methods have their limitations with respect to increasing larger problems. Certainly, they can be transformed into optimization-based heuristics which can solve larger problems. However, when it comes to huge instances, heuristics are still the answer. Laporte also observed that over time the research community has designed metaheuristics that have become more and more over-engineered at the expense of computation time. He suggests researchers should consider producing simpler and more flexible algorithms capable of faster handling of a broader variety of constraints, even if they cause a slight decrease algorithmic effectiveness.

The application of OR/MS methods in this area has lead to significant achievements in practice. Kant et al. (2008) report on a very successful implementation undertaken by Coca-Cola Enterprises (CCE), the world's largest bottler and distributor of Coca-Cola products. The CCE fleet in the U.S. is only surpassed in size by that of the U.S. Postal Service. The software developed is very flexible and handles a variety of practical constraints in determining the truck routes from each distribution center to the retail outlets. Hundreds of dispatchers use this software daily to plan the routes for tens of thousands of trucks. The deployment of the software has resulted in annual cost savings of tens of millions of dollars. In addition, CCE has experienced fewer missed deliveries and gained the ability to deal with tighter time windows, thereby substantially enhancing its customer service. Given the success of the software, Coca-Cola decided to roll it out in other parts of its business.

A variety of routing settings also involve the temporal aspect in the form of customer imposed time windows. A unified framework for all time constrained vehicle routing and crew scheduling problems was developed by Desaulniers et al. (1998). This paper presents a more general model than previously considered which integrates all the different time constrained vehicle routing and crew scheduling problem types examined up that point in the literature. The model extends well-known generic formulations to allow the modeling of all real-world circumstances encountered to date in this environment. This enables the reader to understand the common structure of these problems. It also allows one to perceive the relations between the various problems, the different forms of the model used previously in the literature, and assorted applications across a unified formulation. This also permits the reader to note the diversity of specialized algorithms that have been designed to solve them, and to comprehend the difficulties inherent in certain modeling aspects.

The common structure of these problems is a multi-commodity network flow model with additional resource constraints. Time is one example of a resource. Resource variables help manage complex nonlinear cost functions and difficult local constraints (e.g., time windows, vehicle capacity, and union rules). To solve the nonlinear multi-commodity problems in this class, the paper presents a branch-and-bound framework. It shows that a variety of strategies and algorithms can be utilized for the computation of lower bounds and for devising branching schemes. The lower bounds are derived by using a decomposition approach. In their paper, Desaulniers et al. focus on an extension of the Dantzig-Wolfe decomposition principle and establish that this is valid even for nonlinear objective functions and constraints.

They also illustrate that it embeds the column generation-based methods using set partitioning formulations previously suggested in the literature as special cases. The branching module used to obtain integer solutions compatible with column generation is more general, but yet simpler than other prior strategies. Branching decisions and cuts appear either in the master problem or in the subproblem structures. Finally, the authors examine the constrained shortest path problems that appear at the subproblem level of the decomposition. The paper displays the variety of specialized dynamic programming algorithms that have been developed to solve these and more general single commodity problems and the aspects which have not yet received attention.

Optimal algorithms stemming from the above framework have emerged as the most preferred solution methodologies. These branch, price, and cut algorithms have been widely applied not only in a variety of routing and scheduling transportation contexts, but also in crew scheduling, network design, production, and telecommunications, as well as other areas. These algorithms have become even more powerful due to different classes of strong cutting planes that have been proposed to tighten the lower bounds. Significant improvements in the quality of the lower bounds computed in the search tree have also resulted from utilizing the elementary shortest path problem with resource constraints at the subproblem level.

Crew Scheduling

Two notable application areas of the above framework are the urban transit crew scheduling problem and the airline crew scheduling problem. Blais et al. (1990) describe a software package to handle the former problem. It consists of several modules. The first uses standard network flow methodology to solve the bus scheduling problem. Next, crew scheduling is handled in two steps. In the first, several approximations are used to permit the fast derivation of a linear- programming solution. Using this solution, specific driver assignments are then obtained in step two by means of solving a quadratic-integer program heuristically and using an optimal matching algorithm. Finally, a shortest path algorithm utilizing the marginal costs from the matching problem is used to improve the solution. The software has been successfully implemented in a number of cities worldwide.

With respect to exact algorithms, very large multiple-depot vehicle scheduling problems can be solved to optimality in reasonable times. The same holds true for practical crew scheduling problems encountered in urban mass transit and in air transportation. However, the joint consideration of these two problems proved to be much more challenging. Haase et al. (2001) address this simultaneous vehicle and crew scheduling problem in urban mass transit systems. They propose an optimization algorithm based on the above Dantzig-Wolfe column-generation framework for the problem variant involving a single depot case and a homogeneous vehicle fleet. The authors take a crew-first, vehicle-second approach where decision variables are defined only for the scheduling of drivers. The bus routes are handled within constraints. These constraints ensure that optimal bus itineraries can be obtained in polynomial time once the crews have been scheduled. The authors provide computational results that indicate that this technique was capable to optimally solve larger problems than previously reported in the literature. An easily achieved optimization-based heuristic version of the method is was able to solve even larger instances.

The evolution in airline crew scheduling from the manual methods of the early 1970s to the powerful OR/MS based software now in use mirrors the developments that have occurred in many other logistics areas. In addition, research in crew scheduling is part of the stream of research spearheading the development of optimization methods capable of handling practical size problems. This new generation of optimal algorithms discussed above blends the effectiveness of advanced optimization methods, designed to take advantage of special problem structures, with the efficiency of sophisticated computer science techniques, and the computing power of workstations.

Air transport carriers use a five-phase tactical planning and scheduling process. The schedule planning phase first determines all flight segments, or legs, to be flown during a given period, according to the forecasted demand, the time slots that the company owns at different airports, and the competition. The next phase is fleeting, where each equipment type or

fleet is assigned to individual legs. The fleeting solution provides a decomposition for the problems to be considered in the next three phases. For each fleet, the flight legs with their corresponding scheduled departure and arrival times become inputs to the aircraft routing phase. At this stage, for each type of aircraft, routes are build that must encompass all legs to be flown and satisfy maintenance requirements. The fourth phase builds valid crew pairings, also known as crew rotations, to minimize crew cost. A pairing is a detailed schedule of activities, such as flight legs, deadhead legs (crew members fly as passengers), briefings and debriefings, breaks and nighttime rests that start and end at the same crew base. In the fifth phase, employees are assigned to monthly blocks where each block describes the activities of a crew member during the month. When this process accounts for employee preferences it is called rostering. When blocks are built without regard to crew members' desires, the process is called bidding, in which case, crew members choose blocks according to seniority.

Butchers et al. (2001) provide a historical account and discuss the OR/MS techniques developed for crew scheduling and rostering at a major airline over a fifteen year period. It highlights the fact that the use of such methodologies created major savings for the company, while at the same time providing rosters that benefited the crew members. The account is also illustrative of the advantages to be derived from close collaborations between industry and academia. Nevertheless, the airline planning process phases considered had to be treated sequentially due to the size of the problems involved. The fact that in this planning process the output of an earlier phase provides the input to the next later phase generally leads to suboptimal policies.

Researchers have started to solve selected subsets of planning problems such as fleeting and aircraft routing and aircraft routing and crew pairing simultaneously. Representative of this line of work is that of Sandhu and Klabjan (2007) that addresses the fleeting, aircraft routing, and crew pairing phases in an integrated fashion. The maintenance requirements that must be satisfied in the aircraft routing phase are, however, not considered. The authors propose two optimal algorithms, one using a Benders decomposition approach and the other involving a combination of Lagrangian relaxation and column generation. Based on computational experiments conducted using data from a major carrier, they conclude that if improvements are sought in a short amount of time, the former method should be used. However, if sufficient computing time is available, the usual case in this planning environment, then the latter technique should be utilized. In addition, the authors found the Lagrangian relaxation/column generation approach more robust and practical.

Real-Time Logistics

While the size of problems solved by optimization algorithms increases constantly, heuristics remain a viable tool for very large-scale and/or very complex problems. Dispatching, an intricate activity given the need for a solution in real-time to large-scale problems, lends itself naturally to heuristic solutions. The use of fast route construction/route improvement heuristics to deal with the practical complexities of the problem typifies the kind of research conducted in the 1980s. The highly dynamic character of dispatching is also apparent in truckload transportation. In this environment characterized by high demand uncertainty, a motor carrier must continuously manage the assignment of drivers to loads across the country. Stochastic network optimization models exemplify the type of methodology developed to solve this dynamic vehicle allocation problem. Powell et al. (1995) provide an extensive survey of this problem area.

When shipments could not be forecasted with accuracy, Moore et al. (1991) report having built mixed-integer programming (MIP) and simulation models. The use of these techniques for operational purposes has stemmed from the successful solution of a strategic decision through similar methods. This decision involved the significant reduction in the number of carriers used and the creation of partnerships with them. To solve the carrier selection problem for a global, integrated aluminum company, the authors developed an MIP and further analyzed its results using simulation. This problem represented an important part of a redesign effort aimed at centralizing previously decentralized transportation and purchasing decisions. In particular, by creating a central dispatch center and supporting decisions with OR/MS methodologies, the company improved on time delivery and reduced annual freight costs by

millions of dollars. Overall, this implementation was a reflection of the lean manufacturing philosophy extended to logistics. Furthermore, as logistics has evolved into an information technology centric environment, partnerships with carriers now involve electronic data interchange and web based information sharing.

Supply chains have become a competitive weapon in the global economy. The remarkable advances in telecommunications and information technology have enabled companies to focus on velocity and timeliness throughout the supply chain. To achieve these competitive advantages, they must be able to make effective use of the vast amount of real-time information now available to them. The Dynamic Vehicle Routing Problem (DVRP) is a prime example of a distribution context where intelligent use of real-time information can differentiate one company from another by means of superior on-time service. The DVRP is the dynamic counterpart of the VRP. In the latter problem, the objective is generally to minimize the travel cost for several vehicles that must visit and service a number of customers. Constraints specifying capacity restrictions, time windows within which to start service at customers, and additional requirements on the drivers and vehicles restrict the optimization space. In the VRP all routing and demand information is known with certainty prior to the day of operations, so routes can be planned ahead. In contrast, in the DVRP part or all of the necessary information becomes available only during the day of operation. In other words, not all information relevant to the planning of the routes is known by the planner when the routing process begins and information can change after the initial routes have been constructed.

The practical significance of the DVRP is highlighted by the variety of environments it can model. An important application is the pickup and delivery of overnight mail. Other scenarios include the distribution of heating oil or liquid gas to private households, residential utility repair services, such as cable and telephone, and appliance repair. Additional settings are the transportation of the elderly and physically disabled, taxi cab services, and emergency services, such as police, fire, and ambulance dispatching.

Gendreau and Potvin (2004) have edited a special issue of *Transportation Science* dealing with many issues in real-time fleet management. These were created by the consideration of transportation and fleet management activities as an integral part of the supply chain, their coordination with other aspects of the supply chain, and the explosive growth of web-based logistics services. The paper by Larsen et al. (2004) is illustrative of this type of research. The authors examine the traveling salesman problem with time windows for various degrees of dynamism. The objective is to minimize lateness and examine the impact of this criterion choice on the distance traveled. The focus on lateness is motivated by the problem faced by overnight mail service providers. A real-time solution method is proposed that requires the vehicle, when idle, to wait at the current customer location until it can service another customer without being early. In addition, the authors develop several enhanced versions of this method that may reposition the vehicle at a location different from that of the current customer based on a priori information on future requests. The results obtained on both randomly generated data and on a real-world case study indicate that all policies proved capable of significantly reducing lateness. The results also show that this can be accomplished with only small distance increases.

Another important setting for the application of OR/MS methodologies to support real-time decisions is in the airline industry. Airlines must build aircraft routes and crew rotations to provide scheduled service while maximizing profits. This objective must be achieved in an environment that is difficult to predict. Hence, planning decisions– made in advance– may have to be altered by real-time decisions when perturbations occur in order to minimize customer inconvenience and costs to the airline. Changes made on the day of operations result from bad weather conditions, headwinds on route, technical difficulties with aircraft, crew and passenger delays, and peak-hour congestion at airports. This challenging problem is very important in practice since perturbations are costly in terms of rescheduling issues and especially in terms of loss of traveler goodwill. This is because they can lead to delaying or canceling flights, swapping aircraft among flights or using spare aircraft (if any exist), which in turn affect future deployment of aircraft and crews. Dispatchers usually adjust the planed schedules as soon as a perturbation occurs. They have little time to analyze cost-effective scheduling alternatives. Therefore, it is important to find a good balance between the optimality of a proposed solution and the speed with which it is obtained.

Historically, the day of operations solutions have relied mainly on management information systems and graphical user interfaces, and on simple heuristics to support the decision process. Exact algorithms also have been deployed in practice to provide optimal or near-optimal solutions. Yu et al. (2003) present an optimization based decision support system developed for a large air carrier that provides crew-recovery solutions. The software proved capable of handling major disruptions and in turn it allowed the airline to recover quickly and derive benefits in the millions of dollars.

Inventory in the Supply Chain

The fundamental and often complex trade-offs between transportation and inventory costs are a central issue in supply chain management. Blumenfeld et al. (1987) present an ingenious analysis of the production network of a manufacturer of vehicle components. Their bottom-up approach begins with the analysis of the trade-offs on a single link. These are obtained using a standard economic order quantity (EOQ) model. Using several realistic approximations, the authors are then able to extend their analysis to much more complex networks. In particular, one approximation allows the decomposition of a large network into a number of small independent subnetworks, where shipment sizes can be computed using the single link model. This work involving simple, easy to understand models, supplemented by insightful graphical information, is representative of a line of research complementary to combinatorial optimization.

In light of intense global competitive pressures, many companies have tried to decrease their inventory investment while maintaining or improving customer service in their vital business processes. Yet, the implementation of lean manufacturing has lead to significant increases in product variety. In turn, this has augmented the complexity of the after-sales service logistics networks. Cohen et al. (1990) describe the design of a spare parts inventory control system capable of supporting multiple service levels. The building block of their approach is a periodic review, stochastic model for the one-part, one-location case. This model is then extended to a multi-product, one-location case, called the service allocation problem. This is solved using a greedy heuristic. A decomposition approach is utilized for the overall multi-product, multi-echelon problem. It involves a bottom-up procedure which begins by solving the service allocation problems at the lowest echelon. The solutions are then used to deal with the next higher echelon. The algorithm proceeds in this fashion, level-by-level up to the highest echelon. The model has been implemented by a global computer manufacturer. It has found applicability both as a strategic network redesign tool and as a weekly operational device.

Inventory investment becomes progressively more substantive with increases in the size of companies holding it. While enterprise resource planning software has provided much needed inventory visibility in the supply chain, these systems do not optimize inventory levels. OR/MS methods do, but as they have become increasingly sophisticated over time, the scale and complexity of supply chains has also augmented. The paper by Farasyn et al. (2011) is representative of these issues. It discusses the implementation of various inventory management solutions at Procter and Gamble (P&G). Given the company has 500 different supply chains, it chose a two pronged approach to realize improvements in inventory levels. P&G first focused on the wide-ranging use of spreadsheet-based inventory models throughout its supply chains. This part of the implementation involved four methods that can locally optimize different parts of the supply chains. The next step dealt with the deployment of integrated multi-echelon inventory software in the company's more complex supply chains. The use of OR/MS technologies led to savings of $1.5 billion in 2009, while service levels were maintained or increased. The authors also highlight the fact that this successful implementation did not rely on tools alone. A buy-in from the various entities involved was of equal importance and so was the fit between the necessities of a business unit and the inventory techniques it will use.

Supply Chain Management

Corporations have evolved from the vertical management of separate individual functions to the horizontal management across all functions. Many of

the old conflicts among business units, including transportation versus inventory have given way to the concept of the total logistics cost. Supply chain management is the natural progression of applying these concepts throughout distribution channels by means of pipeline inventory management and information sharing by all involved parties.

The implementation of a comprehensive set of OR/MS tools in a variety of business areas of a large oil company is discussed in Klingman et al. (1987). It is not surprising to see that this industry was at the leading edge of computer integrated horizontal management across functional areas. OR/MS techniques such as linear programming have been utilized in the oil industry since the 1950s. The work of the above authors included such tools as mathematical programming, statistics, forecasting, expert systems, artificial intelligence, organizational theory, cognitive psychology and information systems. A core element was the optimization-based integrated system for supply, distribution, and marketing. This strategic tool is used to make a number of decisions including how much product to buy or trade, how much to hold in inventory, and how much to ship by each mode of transportation. The system is based on the minimum-cost flow network model.

Since then, supply-chain management has become a key application area for OR/MS methodologies, with an explosive growth in the development of models and algorithms and their implementation. Some researchers took an economics perspective, including game theory and information management approaches, while others examined inventory models. Supply chain configuration has also been at the forefront of research in this area. Researchers have examined the integration and coordination between production and distribution, location and routing, routing and inventory, and routing and crew scheduling. They have proposed a vast assortment of heuristic and optimal methods for these aspects of supply chains and a variety of single and multi-objective decision support systems for the overall system design, (Simchi-Levi et al. 2004).

Sophisticated OR/MS models and algorithms are only part of successful implementations. Ulstein et al. (2006) drive home the idea of the collaboration between business and academia, and business and the community as additional necessary ingredients. Their work was conducted for Elkem's silicon division which is the largest supplier of silicon metal and ferrosilicon in the world. With the slowdown in the global economy that started in 2000, the corporation realized the need to improve the efficiency of its supply chain network and evaluate its product portfolio. To help the division to manage this process, the authors developed a strategic planning model. This mathematical-programming model addresses decisions pertaining to future plant structure, including possible closures, new plant acquisitions, and investments in production equipment. The silicon division has used the model and its scenario analysis capabilities extensively to obtain important benefits. The company agreed to a restructuring process, that included reopening a closed furnace and investing $17 million in equipment conversion. Overall, as a result of the restructuring plan, Elkem has achieved significant and sustained improvements in yearly revenue for the silicon division. Many companies face supply-chain design problems with a similar level of complexity. They can benefit from following the close collaborative process described in this paper and from using optimization tools to solve their decision problems.

Sustainability issues are becoming a requisite part of a supply chain studies. For example, Nagurney and Nagurney (2010) consider a company's multicriteria decision problem that attempts to minimize the total costs associated with its supply chain activities, along with the emissions generated by its manufacturing, storage and distribution facets. The business incurs both capital and operational costs. The authors propose a network optimization framework and illustrate an algorithm applied to a number of sustainable supply chain examples. Carter and Easton (2011) trace the evolution of the field from the original research on social and environmental areas, to issues of corporate social responsibility, and the eventual realization that sustainability is part of the bottom line. They provide a comprehensive review of the sustainable supply-chain management literature. One of the salient features of the paper is the relationship between supply chain risk management and contingency planning and sustainable supply chains.

See

- Airline Industry Operations Research
- Crew Scheduling
- Facility Location
- Inventory Modeling
- Multicommodity Network Flows
- Network
- Scheduling and Sequencing
- Supply Chain Management
- Vehicle Routing

References

Blais, J. Y., Lamont, J., & Rousseau, J.-M. (1990). The HASTUS vehicle and manpower scheduling system at the societé de transport de la communauté urbaine de Montréal. *Interfaces, 20*(1), 26–42.

Blumenfeld, D., et al. (1987). Reducing logistics costs at general motors. *Interfaces, 17*(1), 26–47.

Braklow, J., et al. (1992). Interactive optimization improves service and performance for yellow freight system. *Interfaces, 22*(1), 147–172.

Butchers, E., et al. (2001). Optimized crew scheduling at Air New Zealand. *Interfaces, 31*(1), 30–56.

Carter, C., & Easton, P. (2011). Sustainable supply chain management: Evolution and future directions. *International Journal of Physical Distribution and Logistics Management, 41*(1), 46–62.

Cohen, M., et al. (1990). Optimizer: IBM's multi-echelon inventory system for managing service logistics. *Interfaces, 20*(1), 65–82.

Desaulniers, G., et al. (1998). A unified framework for deterministic time constrained vehicle routing and crew scheduling problems. In T. Crainic & G. Laporte (Eds.), *Fleet management and logistics* (pp. 57–93). Norwell, MA: Kluwer.

Farasyn, I., et al. (2011). Inventory optimization at Procter & Gamble: Achieving real benefits through user adoption of inventory tools. *Interfaces, 41*(1), 66–78.

Gendreau, M., & Potvin, J.-Y. (2004). Issues in real-time fleet management. *Transportation Science, 38*(4), 397–398.

Golden, B., Raghavan, S., & Wasil, E. (Eds.). (2008). *The vehicle routing problem: Latest advances and new challenges*. New York: Springer.

Guide, D., & Van Wassenhove, L. (2009). The evolution of closed-loop supply chain research. *Operations Research, 57*(1), 10–18.

Haase, K., Desaulniers, G., & Desrosiers, J. (2001). Simultaneous vehicle and crew scheduling in urban mass transit systems. *Transportation Science, 35*(3), 286–303.

Johnson, M. (2006). Supply chain management: Technology, globalization, and policy at a crossroads. *Interfaces, 36*(3), 191–193.

Kant, G., Jacks, M., & Aantjes, C. (2008). Coca-cola enterprises optimizes vehicle routes for efficient product delivery. *Interfaces, 38*(1), 40–50.

Klingman, D., et al. (1987). The successful deployment of management science throughout citgo petroleum corporation. *Interfaces, 17*(1), 4–25.

Laporte, G. (2009). Fifty years of vehicle routing. *Transportation Science, 43*(4), 408–416.

Larsen, A., Madsen, O., & Solomon, M. M. (2004). The a priori dynamic traveling salesman problem with time windows. *Transportation Science, 38*(4), 459–472.

Moore, E., Warmke, J., & Gorban, L. (1991). The indispensable role of management science in centralizing freight operations at Reynolds metals company. *Interfaces, 21*(1), 107–129.

Nagurney, A., & Nagurney, L. (2010). Sustainable supply chain network design: A multicriteria perspective. *International Journal of Sustainable Engineering, 3*(3), 189–197.

Powell, W., Jaillet, P., & Odoni, A. (1995). Stochastic and dynamic routing and networks. In M. Ball et al. (Eds.), *Handbooks in operations research/management science* (Vol. 8, pp. 141–295). Amsterdam, The Netherlands: Elsevier.

Sandhu, R., & Klabjan, D. (2007). Integrated airline fleeting and crew-pairing decisions. *Operations Research, 55*(3), 439–456.

Simão, H., George, A., & Powell, W. (2010). Approximate dynamic programming captures fleet operations for Schneider national. *Interfaces, 40*(5), 342–352.

Simchi-Levi, D., Chen, X., & Bramel, J. (2004). *The logic of logistics: Theory, algorithms, and applications for logistics and supply chain management* (2nd ed.). New York: Springer.

Toth, P., & Vigo, D. (eds) (2002). The vehicle routing problem. *SIAM Monographs on discrete mathematics and applications, society for industrial and applied mathematics*, Philadelphia.

Ulstein, N., et al. (2006). Elkem uses optimization in redesigning its supply chain. *Interfaces, 36*(4), 314–325.

Yu, G., et al. (2003). A new era for crew recovery at continental airlines. *Interfaces, 33*(1), 5–22.

Log-Linear Model

- Learning Curves
- Regression Analysis

Longest-Route Problem

In a directed network, the finding of the longest route between two nodes is the longest-route problem. In an acyclic network, one that represents the precedence

relationships between activities in a project, the longest route in the network represents the critical path, with the value of the longest route equal to the value of the earliest completion time of the project.

See

- ▶ Critical Path Method (CPM)
- ▶ Program Evaluation and Review Technique (PERT)

Long-Tailed Distribution

- ▶ Heavy-Tailed Distribution

Loss Function

- ▶ Decision Analysis
- ▶ Total Quality Management

Lottery

In utility theory and decision analysis, a lottery consists of a finite number of alternatives of prizes $A_1 \ldots A_n$ and a chance mechanism such that prize A_i will be an outcome of the random experiment with probability $p_i \geq 0, \Sigma_i p_i = 1$.

See

- ▶ Decision Analysis
- ▶ Utility Theory

Lower-Bounded Variables

The condition $l_j \leq x_j$, $l_j \neq 0$, defines x_j as a lower-bounded variable. Such conditions are often part of the constraint set of an optimization problem. For linear programming, these conditions can be removed explicitly by appropriate transformations, given that the problem is feasible when $x_j = l_j$ for each j.

Lowest Index Anticycling Rules

- ▶ Bland's Anticycling Rules

LP

- ▶ Linear Programming

LU matrix decomposition

The decomposition of a matrix into the product of a lower- and an upper-triangular matrix. This is similar to an $\boldsymbol{LDU}$ decomposition in which the $\boldsymbol{D}$ and $\boldsymbol{U}$ matrices have been combined.

See

- ▶ LDU Matrix Decomposition