F

Facilities Layout Problems

FLP

L. R. FOULDS[1], HORST W. HAMACHER[2]
[1] Department Management Systems,
University Waikato, Waikato, New Zealand
[2] Fachbereich Math., University Kaiserslautern,
Kaiserslautern, Germany

MSC2000: 90B80

Article Outline

Keywords
See also
References

Keywords

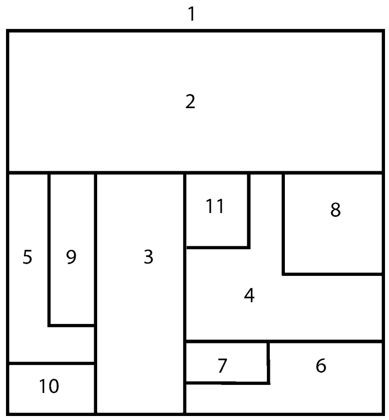
Multifacilities location; REL chart scores; NP-hard; Heuristics; Improved procedure; Construction procedures; Quadratic assignment problem; Location; Graph theory; Graph; Planar subgraph; Adjacency graph; Integer programming; Lagrangian relaxation; Bounds; Branch and bound; Simulated annealing; Genetic algorithms; Multicriteria objective function; Transportation cost; Decision support system; Layout manager

Facilities layout (FL) is concerned with the placement, relative to one another, of the facilities of some physical system. This area differs from planar multifacilities location (MFL; cf. also ► Multifacility and Restricted Location Problems) in that in FL the facilities are all assumed to have a significant physical area and are to be placed in a finite total area which represents their physical system. In MFL the facilities are assumed to be dimensionless points.

The aim of FL is to produce a scale plan (in some scenarios called a *block plan*) of the physical system to be designed. The plan depicts the facilities of the system (each one having its given area and shape) laid out relative to each other. An example of a simple block plan is shown in Fig. 1.

The identification of effective plans depends upon interfacility relationships, which may be quantitative (e. g. transportation costs) or qualitative (e. g. utility scores, called *REL chart scores*, based on facility adjacency). Each FL problem involves optimizing one or more objective functions based on the given interfacility relationship.

FL is an important application area of optimization. This is partly because increased global competition in manufacturing has spurred renewed efforts to



Facilities Layout Problems, Figure 1
A block plan with 11 facilities, including the exterior region, indicated as facility 1

reduce production costs. Efficient physical layout design of manufacturing plants is critical in the quest to achieve and maintain competitive productivity. Indeed, up to 70% of the operating costs of a manufacturing system are related to materials handling and layout. This is because improved layout design often brings about reductions in materials handling, transportation, congestion, and work-in-process.

There are applications of FL techniques in areas other than manufacturing plant design. Examples include the design of office blocks and other commercial buildings, hospitals and other public services, and university campuses, government agencies, and sports complexes. As will become evident in the following discussion, most FL models are *NP*-hard in the strong sense (cf. also ► Complexity Theory; ► Complexity Classes in Optimization). This has reinforced the search for effective heuristics for them.

One of the earliest and best-known FL heuristics is termed CRAFT (coordinate relative allocation of facilities technique) [1]. It requires an initial block plan as input, which it attempts to improve by exchanging the positions of two or three facilities at a time. In contrast to this improved procedure, many other early FL heuristics are construction procedures which build up the final block plan iteratively, by placing facilities sequentially. The serial decision process requires, at each step:

i) a selection of which facility is to be placed next in the block plan being constructed, and
ii) a decision as to where this facility is going to be placed.

Early construction procedures include: CORELLAP [23] and ALDEP [32].

One of the major FL optimization models is based on the quadratic assignment problem (QAP; cf. also ► Quadratic Assignment Problem). For overviews on this subject see [4,5,29]. Formulations of various FL problems based on the QAP involve minimizing the total transportation cost between all pairs of facilities. This total cost comprises a sum of components calculated according to the distance and the amount of work flow between each pair of facilities. The constraints of the QAP model are based on the assumption that the block plan is tessellated into a grid of unit squares (called *locations*) and that no two facilities are to be assigned the same location. Many of these models assume that all the facilities are of equal area. However, when facilities have unequal areas or irregular shapes, additional constraints must be added. The facilities are partitioned into a number of subfacilities of unit area. The problem then is to locate the subfacilities so that all the subfacilities of each facility are assigned adjacent locations in an appropriate configuration. As the QAP is *NP*-hard, most FL applications of it are concerned with heuristics. A QAP model of a common FL problem is:

$$\begin{cases} \min & \displaystyle\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{r=1}^{n} a_{ijkr}x_{ij}x_{kr} \\ \text{s.t.} & \displaystyle\sum_{i=1}^{n} x_{ij} = 1, \quad j = 1,\dots,n, \\ & \displaystyle\sum_{j=1}^{n} x_{ij} = 1, \quad i = 1,\dots,n, \\ & x_{ij} = 0 \text{ or } 1, \quad i,j = 1,\dots,n, \end{cases}$$

where

n = the number of subfacilities,

c_{ij} = the cost per unit time period of assigning subfacility i to location j. (These costs are usually one-time relocation costs which are converted to an annual equivalent.)

d_{jr} = the cost per movement or interaction over the distance from location i to location r,

f_{ik} = the number of moves per time period in the workflow from subfacility i to subfacility k,

S_i = the set of locations to which subfacility i may be feasibly assigned,

$$a_{ijkr} = \begin{cases} f_{ik}d_{jr} & \text{if } i \neq k \text{ or } j \neq r, \\ c_{ij} & \text{if } i = k \text{ and } j = r, \end{cases}$$

$$x_{ij} = \begin{cases} 1 & \text{if subfacility } i \\ & \quad \text{is assigned to location } j, \\ 0 & \text{otherwise.} \end{cases}$$

If there are more locations than subfacilities, a number of dummy facilities can be introduced with zero c_{ij} and f_{ik} values. The f_{ik} values are set to relatively high levels if subfacilities i and k belong to the same facility, thereby ensuring their adjacency. The c_{ij} values are set to relatively high values when $j \notin S_i$.

A second major FL optimization model is based on graph theory (GT) and involves maximizing the sum of the REL chart scores corresponding to the pairs of facilities that are adjacent in the block plan. The formulations can accommodate specifications that the region exterior to the block plan is one of the facilities, and that the facilities are of unequal areas and various shapes. GT models represent facilities and the possible adjacency of pairs of facilities in the block plan by the vertices and edges of a graph, respectively. The REL chart scores are used to weight the edges of the graph. The objective is to identify the planar subgraph (termed an *adjacency graph*) of this graph with the largest total weight in terms of its REL chart scores. The optimal adjacency graph specifies which pairs of facilities are to be placed adjacent to each other in the block plan. As this model was shown to be *NP*-hard in [13], most research concentrates on heuristics. However, some GT algorithms for FL problems guaranteeing optimality, do exist. The algorithm in [12] involves a series of tests for determining whether a proposed adjacency graph being constructed is planar or not. In [6] an integer programming formulation based on the GT approach is discussed. It employs a Lagrangian relaxation procedure (cf. also ► Integer Programming: Lagrangian Relaxation) for the derivation of bounds to be used in a branch and bound algorithm (cf. also ► Integer Programming: Branch and Bound Methods). Approaches to enforce connectivity of subgraphs corresponding to facilities are taken from *k*-cardinality tree models ([14] and [7]) which can also incorporate forbidden areas [10,11].

Early GT heuristics first identify the adjacency graph and then attempt to construct a block plan corresponding to the information provided by the graph. Examples include the heuristics of [3,9,24] and [27]. The comparisons in [28] show that the results of [27] are invariably so close to optimality that the quest for heuristics which find good quality adjacency graphs can now be considered essentially solved. More recent GT heuristics build up the adjacency graph and its corresponding block plan simultaneously, such as the heuristics of [37].

It has been observed that many of the previously mentioned techniques are not computationally feasible for some of the large scale numerical instances of FL problems encountered in industry and often identify local optima which are clearly far from globally optimal. This has given rise to many investigations into whether the more recently developed random search procedures (such as simulated annealing (SA; cf. also ► Simulated Annealing Methods in Protein Folding) and genetic algorithms (GA; cf. also ► Genetic Algorithms)) could be used to devise useful FL heuristics. There is a fundamental difference between SA and GA. That is, GA must, of necessity, deal with a set of possible solutions to the problem in hand, while SA considers only one possible solution at a time. Because GA explores the set of all feasible solutions by combining the characteristics of various single feasible solutions, it sometimes covers a larger portion of the solution space than SA, within the same computational time. Thus, it appears to be the more successful of the two for FL problems.

SA can be applied to FL problems in a variety of ways. There exist SA improvement heuristics for FL problems with

i) multiple floors, (based on the improvement approach) [26],
ii) multiple objectives based on both transportation costs and REL chart scores [33].

For further information see [16] and [22]. However, it appears that the logarithmic cooling schedule of SA causes its FL heuristics to perform relatively slowly. For this reason it seems that GA heuristics are more effective for FL problems. For instance the GA approach to solve the QAP, devised in [35], can be applied to QAP models of FL problems, such as the one given earlier. However, this GA heuristic has only a single solution giving rise to a mutant, which means that parallelism is lost to a certain extent.

To overcome this deficiency, it is possible to design more effective GA heuristics for FL problems by adopting a small mutation rate and a large crossover rate. A heuristic with efficient crossover operators with low level mutation has been devised in [34]. Further heuristic attempts to tackle the QAP include tabu search (see e. g. [2]) or the reverse elimination method [36].

The approaches to FL described so far have been classical in the sense that they have nearly all embraced single objective functions. In contrast, there have been developments in FL models with multiple criteria. Examples include: a multifactor plant layout methodology devised in [15], a layout planning system with multiple criteria and a variable domain representation in [18], an expert system using priorities for solving multiple criteria facilities layout problems in [25], and a multiattribute decision theoretic approach for layout design in [31].

There are numerous computer programs in existence which implement FL heuristics. Three early ones from the 1960s: CRAFT, CORELAP, and ALDEP have already been discussed. In the 1970s two improvement-style heuristics, both based on CRAFT, appeared to be among the best of those proposed then. FRAT (facilities relative allocation technique) [21] assumes that all the facilities have equal areas. TSP (terminal sampling procedure) [17] carries out the interchange of the placement, in the block plan, of pairs of facilities on a selective basis. The program has the ability to use improved block plans as input and to fix the placement of certain facilities. Three of the large number of FL programs written in the 1980s will be mentioned. SPACECRAFT [20] is an extension of CRAFT to multifloor FL problems. See [17] for a perturbation scheme, and [18] for a new FL system which accommodates a variety of types of spaces, including solid, circulation, and empty. A multicriteria objective function involves transportation cost, REL chart scores, the percentage of unused area, and block plan structure.

The 1990s saw a different type of FL program emerging: the *decision support system* (DSS). One such example, called *layout manager* [8] is a user-friendly menu-driven DSS which provides for the choice between a number of optimality criteria including, among others, transportation cost and REL chart scores. The system is written in Pascal, within the Microsoft Windows environment.

See also

- ▶ Combinatorial Optimization Algorithms in Resource Allocation Problems
- ▶ Competitive Facility Location
- ▶ Facility Location with Externalities
- ▶ Facility Location Problems with Spatial Interaction
- ▶ Facility Location with Staircase Costs
- ▶ Global Optimization in Weber's Problem with Attraction and Repulsion
- ▶ MINLP: Application in Facility Location-allocation
- ▶ Multifacility and Restricted Location Problems
- ▶ Network Location: Covering Problems
- ▶ Optimizing Facility Location with Rectilinear Distances
- ▶ Production-distribution System Design Problem
- ▶ Resource Allocation for Epidemic Control
- ▶ Single Facility Location: Circle Covering Problem
- ▶ Single Facility Location: Multi-objective Euclidean Distance Location
- ▶ Single Facility Location: Multi-objective Rectilinear Distance Location
- ▶ Stochastic Transportation and Location Problems
- ▶ Voronoi Diagrams in Facility Location
- ▶ Warehouse Location Problem

References

1. Armour G, Buffa E (1963) A heuristic algorithm and simulation approach to relative location of facilities. Managem Sci A:294–309
2. Battiti R, Tecciolli G (1992) Parallel biased search for combinatorial optimization: Genetic algorithms and tabu search. Microprocessors and Microsystems 16:351–367
3. Boswell SG (1992) A new greedy heuristic for facilities planning. Internat J Production Res 30:1957–1968
4. Burkard RE (1990) Locations with spatial interactions: The quadratic assignment problem. In: Francis RL, Mirchandani P (eds) Discrete Location Theory. Wiley, New York, pp 387–437
5. Cela E (1998) The quadratic assignment problem: Theory and algorithms. Kluwer, Dordrecht
6. Christofides N, Galliani G, Stefanini L. An algorithm for the maximal planar graph problem based on Lagrangean relaxation. Math Program
7. Fishetti M, Hamacher HW, Jörnsten K, Maffioli F (1994) Weighted k-cardinality trees: Complexity and polyhedral structure. NETWORKS 24:11–21
8. Foulds LR (1997) Layout manager: A microcomputer-based decision support system for facilities layout. Decision Support Systems 20:199–213
9. Foulds LR, Gibbons PB, Giffin JW (1985) Facilities layout adjacency determination: An experimental comparison of three graph theoretic heuristics. Oper Res 33:1091–1106
10. Foulds LR, Hamacher HW (1992) A new integer programming approach to (resticted) facilities layout problems allowing flexible facility shapes. Res Report Dept Management Systems Univ Waikato
11. Foulds LR, Hamacher HW, Wilson JM (1998) Integer programming approaches to facilities layout models with forbidden areas. Ann Oper Res 81:405–417
12. Foulds LR, Robinson DF (1976) A strategy for solving the plant layout problem. Oper Res Quart 27:845–855
13. Giffin JW (1985) Graph theoretic techniques for facilities layout. PhD Thesis, Dept Economics, Univ Canterbury, Christchurch, New Zealand

14. Hamacher HW, Joernsten K, Maffioli F (1993) Weighted k-cardinality trees. Techn Report, Dipt Elettronica, Politecnico Milano, 91.023
15. Harmonosku CM, Tothero GK (1992) A multi-factor plant layout methodology. Internat J Production Res 30:1773–1790
16. Heregu SS, Alfa AS (1992) Experimental analysis of simulated annealing-based algorithms for the layout problem. Europ J Oper Res 57:190–202
17. Hitchings GG, Cottam M (1976) An efficient heuristic procedure for solving the layout design problem. OMEGA Internat J Management Sci 4:205–214
18. Jacobs FR (1987) A layout planning system with multiple criteria and a variable domain representation. Managem Sci 33:1020–1034
19. Jajodia S, Minis I, Harhalakis G, Proth JM (1992) CLASS: Computerised layout solutions using simulated annealing. Internat J Production Res 30:95–108
20. Johnson RV (1982) SPACECRAFT for multi-floor layout planning. Managem Sci 28:407–417
21. Khalil TM (1973) Facilities relative allocation technique (FRAT). Internat J Production Res 11:183–194
22. Kouvelis P, Chiang WC, Fitzsimmons J (1992) Simulated annealing for machine layout problems in the presence of zoning constraints. Europ J Oper Res 57:203–223
23. Lee R, Moore JM (1967) CORELAP – Computerized relationship layout planning. J Industr Eng 18:195–200
24. Leung J (1992) A new graph theoretic heuristic for facility layout. Managem Sci 38:594–605
25. Malakooti B, Tsirisjo A (1989) An expert system using priorities for solving multiple criteria facility layout problems. Internat J Production Res 27:793–808
26. Meller RD, Bozer YA (1986) A new simulated annealing algorithm for the facility layout problem. Internat J Production Res 34(6):1675–1692
27. Merker J, Waescher G (1987) Two new heuristic algorithms for the maximal planar layout problem. OR Spektrum 19:131–137
28. Merker J, Waescher G (1997) A comparative evaluation of heuristics for the adjacency problem in facility layout planning. Internat J Production Res 33:447–466
29. Pardalos PM, Wolkowicz H (1994) Quadratic assignment and related problems. DIMACS, vol 16. Amer Math Soc, Providence
30. Picone CJ, Wilhelm WE (1984) A perturbation scheme to improve Hillier's solution to the facilities layout problem. Managem Sci 30:1238–1249
31. Sarin SC, Loharjun P, Malmborg CJ, Krishnakumar A (1992) A multiattribute decision-theoretic approach for the layout design problem. Europ J Oper Res 57:231–242
32. Seehof JF, Evans WO (1967) Automated layout design problems. J Industr Eng 18:690–695
33. Suresh G, Sahu S (1993) Multi-objective facility layout using simulated annealing. Internat J Production Economics 32:239–254
34. Suresh G, Vinod VV, Sahu S (1995) A genetic algorithm for facility layout. Internat J Production Res 33:3411–3423
35. Tate DM, Smith AE (1995) A genetic approach to the quadratic assignment problem. Comput Oper Res 32:73–83
36. Voss S (1995) Solving quadratic assignment problems using the reverse elimination method. In: Nash SG, Sofer A (eds) The Impact of Emerging Technologies on Computer Sci and Operations Res. Kluwer, Dordrecht
37. Watson K, Giffin JW, Foulds LR (1995) Orthogonal layouts using the deltahedron heuristic. J Australasian Combin Soc 12:127–144

Facility Location with Externalities

MARGARET BRANDEAU
Stanford University, Stanford, USA

MSC2000: 90B80, 90B85, 91Bxx, 90Cxx, 91Axx

Article Outline

Keywords
Location of Mobile Servers
Location of Fixed Service Facilities
System-Optimizing Environment
User-Optimizing Environment
Resource Allocation with Externalities
See also
References

Keywords

Discrete location and assignment; Continuous location; Mathematical economics; Mathematical programming; Game theory

A typical assumption in *facility location* models is that the cost customers face in patronizing facilities is independent of the actions of other customers (with the possible exception of capacity restrictions). For example, many classical facility location models assume that customers patronize the facility (or are served by the facility) that minimizes the cost of travel between the facility and the customer (e. g., see, [12,13]). Other facility location models incorporate marketing considerations, and assume that customers patronize the facility that is 'most attractive' to them, where attractiveness depends not only on travel cost, but also on attributes of each facility such as size, goods offered, and number of servers (e. g., see [15]).

However, in many situations, the cost customers face in patronizing a facility is a function of the actions of other customers. For example, waiting time for service may be longer in a store that is patronized by many customers than in a store with fewer customers. An ambulance that serves a large, heavily-populated area is likely to incur longer delays in providing service than an ambulance serving a smaller, less-populated area. These are examples of negative externalities associated with the market share of the facility. Conversely, in some cases the *externalities* could be positive: for example, a crowded nightclub is likely to be more popular than one that attracts fewer patrons.

If facilities provide essential services (e. g., gasoline, drivers' licenses), customer demand may be constant, regardless of the costs customers face in obtaining services. However, for facilities that provide nonessential services (e. g., fast-food restaurants, retail stores), customer demand might be a function of the total cost of receiving service.

This chapter discusses models for the location of facilities that incorporate not only travel cost but also negative externalities associated with the market share of the facility. Various problem formulations are discussed, and selected references are provided. A more comprehensive discussion is given in [10]. The case of positive externalities is not discussed because, for such problems, degenerate solutions tend to occur (e. g., the optimal solution may be to locate all facilities at the same point, with any point in the region being optimal).

One can consider two different situations regarding the allocation of customer demands to facilities. In a *user-optimizing environment*, customers patronize the facility that minimizes their total cost, in this case travel cost plus externality cost. Such a situation occurs, for example, in customers' selection of grocery stores and bank branches. In a *system-optimizing environment*, customers are assigned to facilities by a central agent. An example is the assignment of voters to polling places.

In the system-optimizing environment, allocation of customer demands can be considered as part of the location optimization problem, similar to many models of facility location that do not incorporate externalities. In the user-optimizing environment, however, models of facility location have at their core a customer-choice *equilibrium* problem: equilibrium occurs when each customer frequents the facility that minimizes his total travel cost plus externality cost. For purely negative externalities, the equilibrium utilization of facilities (total demand satisfied by each facility) is unique, although the equilibrium user-choice pattern (allocation of individual customer demands to facilities) may not be unique ([8,18]). This result holds whether demands are inelastic or elastic with respect to total customer cost. Determination of the user-choice equilibrium can be written as a nonlinear complementarity problem (analogous to [1]), and also as a network flow problem [21] which can be solved using network optimization techniques (e. g., [20]).

This article discusses models for the location of facilities in both types of customer choice environments. A distinction is made between facilities with mobile servers (e. g., ambulances) that travel to fixed customers and return to their home location between calls and facilities that house fixed servers (e. g., postal clerks).

Location of Mobile Servers

Some of the first location models to incorporate externalities were developed in the context of emergency service vehicle location. In such models, the servers (the emergency service vehicles) travel to customers, and the externality cost is the servers' queueing delay. A system-optimizing environment is assumed: customers are assigned to service regions of the servers. Models for determining the home location of such mobile servers have considered a variety of location objectives, including minimization of mean response time to customers (travel time plus queue delay), minimization of the maximum response time to any customer, equalization of server workloads, and other objectives. Examples of such models can be found in [3,4,5,7,11], and [19].

Location of Fixed Service Facilities

Most other facility location models that incorporate externality costs have assumed fixed service facilities.

System-Optimizing Environment

In the system-optimizing environment, since customers are assigned centrally to facilities, it is natural to think only of noncompeting facilities. For the case of fixed customer demands, a natural objective in locating facilities (and allocating customers to facilities) is to minimize total customer cost. This problem is a generalized p-median problem. Such a model might be appropriate for the location of certain public facilities such as voters' polling places. O. Berman and R.C. Larson [2] presented a p-median problem that includes queueing-like congestion of the facilities. In the system-optimizing environment with elastic demands, a natural objective is to locate facilities to maximize the total demand served by facilities (i. e., maximum facility utilization). Such a model might be relevant for the location of fast-food franchises or clinics for preventive childcare (e. g., inoculations). This facility location problem is a generalized p-median problem with an embedded demand equilibrium [18]. For the case of discrete customer demands on a network, S. Kumar [18] proved a nodal optimality theorem and showed that the problem can be formulated as a nonlinear integer convex programming problem and solved using branch and bound (see also [10]).

User-Optimizing Environment

In the user-optimizing environment with fixed service facilities, one can distinguish between noncompeting and competing facilities. For the case of noncompeting facilities in the user-optimizing environment with inelastic demand, a natural location objective is to minimize total customer cost. This framework might be appropriate for the location of public facilities such as Social Security Offices. Assuming discrete customer demands, the problem can be written as a mixed integer bilevel program [10] (given a set of fixed facility locations, one can then determine the user-choice equilibrium utilization of facilities). M.L. Brandeau and S.S. Chiu [8] considered the case of two such facilities on a tree network with nodal demands. They characterized the optimal facility locations, and presented an algorithm for finding those locations.

A typical location objective for the case of competing facilities (whether or not externalities are considered) is maximization of market share. When externalities are not considered, problems of competitive facility location involve a locational equilibrium; when negative externality costs and user-optimizing customer choice are considered, such problems also involve a customer-choice equilibrium. E. Kohlberg [17] considered the location of competing identical facilities on a line with uniformly distributed, inelastic demands where customers select a facility based on the sum of travel time plus waiting time for service. For the case of two facilities, the optimal locations occur at the midpoint of the line, and for the case of more than two facilities, Kohlberg [17] showed that a locational equilibrium does not occur. R.M. Braid [6] analyzed the locational equilibrium for two congested public facilities located by competing governmental jurisdictions in an inelastic-demand environment. Brandeau and Chiu [9] analyzed the case of two competing facilities on a tree network with inelastic demands and a general negative externality function. Such a model might be appropriate for the location of similar competing grocery stores. They assumed a Stackelberg game (with a leader and a follower). They characterized the optimal locations of the leader and the follower, and presented an algorithm for finding those locations.

Kumar [18] considered the location decision of a profit-maximizing firm that locates one facility in a region where a number of competitors are already located and in which customer demand is elastic. An example application is the location of competing retail outlets. The problem is a bilevel programming problem which can be heuristically solved using a gradient projection ascent approach (e. g., [14]).

Resource Allocation with Externalities

If facilities are already located, changing facility locations may be expensive. An alternative is to allocate resources to change the characteristics of the facility (e. g., through training or technological improvements). The question is how to balance the cost of change with the associated benefits (e. g., increased market share, lowered total customer cost). *Resource allocation* problems of this type are discussed in [10] and [16].

See also

- ► Combinatorial Optimization Algorithms in Resource Allocation Problems

- ▶ Competitive Facility Location
- ▶ Facility Location Problems with Spatial Interaction
- ▶ Facility Location with Staircase Costs
- ▶ Global Optimization in Weber's Problem with Attraction and Repulsion
- ▶ MINLP: Application in Facility Location-allocation
- ▶ Multifacility and Restricted Location Problems
- ▶ Network Location: Covering Problems
- ▶ Optimizing Facility Location with Rectilinear Distances
- ▶ Production-distribution System Design Problem
- ▶ Resource Allocation for Epidemic Control
- ▶ Single Facility Location: Circle Covering Problem
- ▶ Single Facility Location: Multi-objective Euclidean Distance Location
- ▶ Single Facility Location: Multi-objective Rectilinear Distance Location
- ▶ Stochastic Transportation and Location Problems
- ▶ Voronoi Diagrams in Facility Location
- ▶ Warehouse Location Problem

References

1. Aashtiani HZ, Magnanti TL (1981) Equilibria on a congested transportation network. SIAM J Alg Discrete Meth 2:213–226
2. Berman O, Larson RC (1982) The median problem with congestion. Comput Oper Res 4:119–126
3. Berman O, Larson RC, Chiu SS (1985) Optimal server location on a network operating as an M/G/1 queue. Oper Res 33:746–771
4. Berman O, Larson RC, Parkan C (1987) The stochastic queue p-median problem. Transport Sci 21:207–216
5. Berman O, Mandowski RR (1986) Location-allocation on congested networks. Europ J Oper Res 26:238–250
6. Braid RM (1991) The locations of congestible facilities in adjacent jurisdictions. Regional Sci and Urban Economics 21:617–626
7. Brandeau ML, Chiu SS (1992) A center location problem with congestion. Ann Oper Res 40:17–32
8. Brandeau ML, Chiu SS (1994) Facility location in a user-optimizing environment with market externalities: Analysis of customer equilibria and optimal facility locations. Location Sci 2:129–147
9. Brandeau ML, Chiu SS (1994) Location of competing private facilities in a user-optimizing environment with market externalities. Transport Sci 28:125–140
10. Brandeau ML, Chiu SS, Kumar S, Grossman TA (1995) Location with market externalities. In: Drezner Z (ed) Facility Location: A Survey of Applications and Methods. Springer, Berlin, pp 121–150
11. Chiu SS, Larson RC (1985) Locating an n-server facility in a stochastic environment. Comput Oper Res 12:509–516
12. Daskin MS (1995) Network and discrete location. Wiley/Interscience, New York
13. Drezner Z (eds) (1995) Facility location: A survey of applications and methods. Springer, Berlin
14. Friesz TL, Tobin RL, Cho HJ, Mehta NJ (1990) Sensitivity analysis based heuristic algorithms for mathematical Programs with variational inequality constraints. Math Program 48:265–284
15. Ghosh A, Craig CS (1983) Formulating retail location strategy in a changing environment. J Marketing 47:56–68
16. Grossman TA (1993) Models for optimal facility utilization decisions. Unpublished PhD Diss., Stanford University
17. Kohlberg E (1983) Equilibrium store locations when consumers minimize travel time plus waiting time. Economics Lett 11:211–216
18. Kumar S (1993) Location decisions for private facilities in the presence of elastic demand and externalities. Unpublished PhD Diss., Stanford University
19. Larson RC (1974) A hypercube queueing model for facility location and redistricting in urban emergency services. Comput Oper Res 1:67–95
20. Tomlin JA (1971) A mathematical programming model for the combined distribution-assignment of traffic. Transport Sci 5:120–140
21. Wardrop JG (1952) Some theoretical aspects of road traffic research. Proc Inst Civil Eng Part II:325–378

Facility Location Problems with Spatial Interaction

KAJ HOLMBERG
Department Math., Linköping Institute Technol.,
Linköping, Sweden

MSC2000: 90B80, 90C10

Article Outline

Keywords
Model
Solution Methods
The Dual Ascent and Adjustment Method
Dual Ascent and Branch-And-Bound
Lagrangian Relaxation
with Subgradient Optimization
Benders Decomposition
Mean Value Cross Decomposition

Comparisons and the Role of Γ
Conclusion
See also
References

Keywords

Location; Spatial interaction; Integer programming; Dual ascent

Facility location problems deal with the question of where to locate certain facilities, so that they can satisfy some kind of demand of a certain set of customers, and so that the total cost is minimized. If the facilities are factories or warehouses and the goods will be shipped from the facilities to the customers, one can assume that the shipments will be made so as to minimize the transportation costs. (See also ► Facility Location with Staircase Costs and ► Stochastic Transportation and Location Problems.) However, if the facilities are hospitals or supermarkets, transportation will consist of customers traveling by their own means to/from the facility, and in such a case, it is not certain that each customer will behave exactly so as to minimize the transportation costs.

So in public facility location problems where the clients are free to make their own choice of facility, one should probably expect different results than those minimizing the transportation costs. Modeling such situations, the objective cannot only be to minimize the total transportation and facility costs. The effect of *spatial interaction* has been used to improve location models of this type. Simple plant location problems with spatial interaction between the travelers have been treated in [3,4,6,19,20,22,23], modeled as a nonlinear, mixed integer programming problem.

In [15] a different model is derived, in a similar way as used in [14], that does not use the approximation yielding entropy terms. The model is called the 'exact' formulation of the simple plant location problem with spatial interaction, because of the usage of the classical way of deriving the gravity model, without doing any approximation.

Assuming integer requirements on the transported amounts enables an exact linearization of the nonlinear costs. This yields a linear, pure zero-one model, to the price of a significantly increased number of variables. Luckily the model has a special structure that can be exploited by several different solution methods.

Model

We now describe a public facility location model, with m possible locations for supply points (plants) and n demand points (client zones). The fixed cost for opening plant i is a_i. At demand point j the demand (the number of clients in zone j) is w_j. Trips will be made between the demand points and the opened plants so that the demand is satisfied. The transportation costs for one trip between plant i and demand point j (i. e. the cost for a client at zone j to get service at plant i) is c_{ij}.

The following variables are introduced.

$$z_i = \begin{cases} 1 & \text{if a plant at location } i \text{ is opened,} \\ 0 & \text{if not,} \end{cases}$$

$$x_{ij} = \begin{cases} \text{the number of trips between} \\ \text{plant } i \text{ and demand point } j \\ \text{(i. e. the number of clients} \\ \text{in zone } j \text{ getting service at plant } i\text{).} \end{cases}$$

The total cost for transportation and opening plants is

$$v_1 = \min \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} + \sum_{i=1}^{m} a_i z_i .$$

As for the spatial interaction, one may note that several *microstates* (obtained by identifying every single client's trip) may yield the same *macrostate* (the x-solution). The macrostate given by the largest number of microstates is the most probable solution, according to [29]. Maximizing the number of microstates yielding x yields another objective function for finding the most likely x-solution.

$$v_2 = \min \sum_{i=1}^{m} \sum_{j=1}^{n} \ln(x_{ij}!).$$

A suitable objective function is now obtained by combining these two parts, $v* = v_2 + \gamma v - 1$, where the weight γ reflects the sensitivity of the system to the costs. For large values of γ, it is very important to minimize the costs, while for smaller values of γ, the costs are not very important.

The best value of the parameter γ, being the weight of how much the clients take the costs into account, must be found by calibrating the model against a real life situation. Considering a certain situation, one can assume that γ is fixed and given.

The following model (SPLPS) is obtained:

$$v^* = \min \sum_{i=1}^{m}\sum_{j=1}^{n} \ln(x_{ij}!) + \gamma\left(\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} + \sum_{i=1}^{m} a_i z_i\right)$$

such that

$$\sum_{i=1}^{m} x_{ij} = w_j, \quad \forall j \tag{1}$$

$$x_{ij} - w_j z_i \le 0, \quad \forall i, j \tag{2}$$

$$x_{ij} \ge 0, \quad \text{integer}, \quad \forall i, j \tag{3}$$

$$z_i \in \{0, 1\}, \quad \forall i. \tag{4}$$

SPLPS is a pure integer problem, with a nonlinear objective function that actually is defined only in the integer points.

If γ is so large that the logarithmic part is negligible, we get pure cost minimization. The model is then identical to the simple plant location problem, SPLP, and can be efficiently solved by for example a dual ascent method, [5].

In previous work, a continuous relaxation of x together with Stirling's approximation, $\ln(x_{ij}!) \approx x_{ij} \ln(x_{ij}) - x_{ij}$, have been used, yielding a nonlinear, mixed integer programming problem.

Now we linearize the cost function for each variable x_{ij} in the interval $0 \le x_{ij} \le w_j$, with break points at each integer point. This does not introduce any error (as Stirling's approximation would). The number of variables then depends on the values of the demands.

We get $\overline{c}_{ijk} = \ln(k!) - \ln((k-1)!) + \gamma c_{ij} = \ln(k) + \gamma c_{ij}$. Note that $\overline{c}_{ijk} > \overline{c}_{ijk-1}$, [18], which indicates convexity of the resulting cost functions.

Then we do the substitution $x_{ij} = \sum_k x_{ijk}$, where x_{ijk} is the amount of x_{ij} that falls in the interval $(k-1, k)$. The following model (SPLPE) is obtained:

$$v^* = \min \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{w_j} \overline{c}_{ijk}x_{ijk} + \sum_{i=1}^{m} \gamma a_i z_i$$

such that

$$\sum_{i=1}^{m}\sum_{k=1}^{w_j} x_{ijk} = w_j, \quad \forall j, \tag{5}$$

$$x_{ijk} - z_i \le 0, \quad \forall i, j, k, \tag{6}$$

$$x_{ijk} \in \{0, 1\}, \quad \forall i, j, k, \tag{7}$$

$$z_i \in \{0, 1\}, \quad \forall i. \tag{8}$$

This is a large linear integer programming problem with $m(1 + \sum_{j=1}^{n} w_j)$ binary variables. The fact that it is a pure 0–1-problem is favorable when it comes to solution methods. The coefficients in the constraints (6) are all reduced to one, so the formulation is probably quite strong.

Solution Methods

It is in principle possible to solve SPLPE with a standard integer programming code, but the size of the model prohibits this for all instances but very small ones. As the model is fairly new, one cannot find many solution methods proposed in the literature.

A dual ascent procedure for this problem has been developed, see [17]. Another method, based on the same dual, is Lagrangian relaxation and subgradient optimization, investigated in [18]. Solution methods based on primal and dual decomposition techniques can also be used, see [16], where one conclusion is that Benders decomposition seems to be an efficient solution method. In [13], the dual ascent approach is inserted in a branch and bound framework, and applied to a somewhat more general problem.

We will briefly describe these methods below.

The Dual Ascent and Adjustment Method

A dual ascent procedure can be used to, in principle, solve the LP-dual of the LP relaxation of SPLPE, by increasing the dual variables in small steps, in such a way that an ascent of the dual function is obtained in each step. Furthermore, a dual adjustment procedure can be used to temporarily decrease dual variables that block further improvement.

Let α_j denote the dual variables corresponding to constraint set 1 of the LP relaxation of SPLPE, β_{ijk} the dual variables corresponding to constraint set 2 and δ_i the dual variables corresponding to the constraints $z_i \leq 1$. The LP-dual will be as follows:

$$v^* = \max \sum_{j=1}^{n} w_j \alpha_j + \sum_{i=1}^{m} \delta_i$$

such that

$$\alpha_j - \beta_{ijk} \leq \overline{c}_{ijk}, \quad \forall i, j, k, \tag{9}$$

$$\sum_{j=1}^{n} \sum_{k=1}^{w_j} \beta_{ijk} - \delta_i \leq \gamma a_i, \quad \forall i, \tag{10}$$

$$\beta_{ijk} \geq 0, \quad \forall i, j, k, \tag{11}$$

$$\delta_i \geq 0, \quad \forall i. \tag{12}$$

The basic steps are to make moves in the dual variables α_j. For fixed $\alpha = \overline{\alpha}$, the LP-dual is trivially solvable, yielding

$$\beta_{ijk} = \max(0, \overline{\alpha}_j - \overline{c}_{ijk}), \quad \forall i, j, k,$$

and

$$\delta_i = \max\left(0, \sum_{j=1}^{n} \sum_{k=1}^{w_j} \beta_{ijk} - \gamma a_i\right), \quad \forall i.$$

Now let k_{ij} be such that

$$\overline{\alpha}_j \geq c_{ijk}, \quad \forall k \leq k_{ij}, \qquad \overline{\alpha}_j < c_{ijk}, \quad \forall k > k_{ij},$$

and

$$q_{ij} = \begin{cases} 1 & \text{if } \overline{\alpha}_j = c_{ijk_{ij}} \\ 0 & \text{if not.} \end{cases}$$

Also, let

$$s_i = \sum_{j=1}^{n} \sum_{k=1}^{w_j} \max(0, \overline{\alpha}_j - c_{ijk})$$

and define $I^{>}=\{i: s_i > \gamma\ a_i\}$, $I^{=}=\{i: s_i = \gamma a_i\}$, $I^{<} = \{i: s_i < \gamma a_i$ and $I^{\geq} = I^{>} \cup I^{=}$.

The complementary slackness conditions are

$$x_{ijk} = z_i, \quad \forall k \leq k_{ij} - q_{ij}, \quad \forall i, j,$$
$$x_{ijk} = 0, \quad \forall k > k_{ij}, \quad \forall i, j,$$
$$z_i = 1, \quad \forall i \in I^{>},$$
$$z_i = 0, \quad \forall i \in I^{<}.$$

Now define

$$w_j^l(\overline{\alpha}) = \sum_{i \in I^{>}} (k_{ij} - q_{ij})$$

and

$$w_j^u(\overline{\alpha}) = \sum_{i \in I^{\geq}} k_{ij}.$$

Then it can be shown, [17], that

$$w_j^l(\overline{\alpha}) \leq \sum_{i=1}^{m} \sum_{k=1}^{w_j} x_{ijk} \leq w_j^u(\overline{\alpha}).$$

This means that w_j^l and w_j^u are lower and upper bounds on the left-hand sides of constraints (9). In order to obtain feasibility (optimality in the dual) the intervals between these bounds should contain the right-hand sides w_j. The following is proved in [17]: If $w_j^l(\overline{\alpha}) \leq w_j \leq w_j^u(\overline{\alpha})$, $\forall j$, then α is optimal in the LP relaxation of SPLPE.

The dual ascent method is now to increase α_j in small steps, so that $w_j^l(\overline{\alpha})$ and $w_j^u(\overline{\alpha})$ increase. The increase of a certain α_j is bounded by the closest breakpoint, induced by the dual constraints of either set 1 (corresponding to enabling or forcing the increase of yet another x_{ijk}) or set 2 (corresponding to enabling or forcing the increase of yet another z_i).

The bounds w_j^l and w_j^u will approach w_j from below, and w_j^l will not be allowed to exceed w_j. The increase of α is repeated, in each step for the j which yields the largest distance between w_j^u and w_j, until optimum is found or improvement is blocked (i. e. a further increase of any α_j would result in $w_j^l > w_j$). In the last case we use an adjustment procedure, which decreases some α_j, in order to allow the increase of other α_j's. Then the ascent phase above is repeated. More details can be found in [17].

Dual Ascent and Branch-And-Bound

The dual ascent and adjustment procedure only solves the LP relaxation of the problem, so to find the exact integer optimum, the procedure must be used within a branch and bound framework. The subproblem in each node of the branch and bound tree is then solved with the dual ascent procedure, in the sense that lower bounds on the optimal objective function value and sometimes feasible primal solutions are obtained. Branches are cut off when the lower bound exceeds the best upper bound known.

One can note that if all the z-variables are fixed in SPLPE, then the problem is trivially solvable, and the x-variables will attain integer values even if the constraints $x_{ijk} \in \{0, 1\}$ are replaced by $0 \leq x_{ijk} \leq 1$, so it may be regarded as an LP-problem. Furthermore it is proved, in [13], that the dual ascent procedure accurately solves the problem when all z-variables are fixed, within a finite number of steps.

Therefore, it is natural to do the branching over the z-variables. Fixed z-variables are handled as follows in the dual ascent phase. Let $I_0 = \{I: z_i \text{ is fixed to } 0\}$ and $I_1 = \{I: z_i \text{ is fixed to } 1\}$. For all $i \in I_0 \cup I_1$, the dual variables δ_i are removed, and the corresponding dual constraints in set 2 are removed. For all $i \in I_1$ the corresponding primal constraints in set 2 are redundant, so we can assume that $\beta_{ijk} = 0$, $\forall\ i \in I_1$, $\forall j, k$. Also, $x_{ijk} = 0$, $\forall\ i \in I_0$, $\forall\ j, k$.

All elements of $I_0 \cup I_1$ must be removed from $I^>$, $I^=$, $I^<$ and $I^{\geq}$. It is not necessary to calculate s_i for $i \in I_0 \cup I_1$. After these changes, the bounds w_j^l and w_j^u are calculated as above.

Some supporting hyperplanes, and breakpoints, are removed from the dual function, as a result of the fixations, so in the dual ascent procedure, fewer steps often need to be taken. (Sometimes the increase of some α_j is limited by the breakpoint where a facility is opened. This will not occur if the facility is fixed open or closed.)

In the worst case, the branch and bound method will enumerate all z-solutions. Thus we have the following result: The dual ascent method within a branch and bound framework will find the exact optimum of SPLPE within a finite number of steps.

In practice branching is done when the dual ascent and adjustment procedure stops, which not necessarily means that the LP-optimum is found. In many cases unnecessary branching is done, and we must expect the branch and bound tree to be larger than it would be for an LP-based branch and bound method.

Branching is done over any z_i, $i \in I^=$, since any value between 0 and 1 is optimal for such a z_i, i. e. the complementary slackness conditions allow for nonintegral values of z_i.

The original dual ascent method starts from zero (no facilities opened and nothing sent). However, for very small values of γ in SPLPS many facilities will be opened, while for very large values of γ the z-solution obtained for the ordinary uncapacitated facility location problem, SPLP, by for examples the dual ascent method DUALOC, [5], might be optimal or close to optimal in SPLPS. In such cases one can use these solutions as starting solutions.

The choice of which dual variable, α_j, to increase first in the dual ascent procedure, could be done cyclically in j, but it seems better to choose the j which exhibits the maximal residual, i. e. the largest gap between w_j^l and w_j.

Lagrangian Relaxation with Subgradient Optimization

Lagrangian relaxation is a well known and often used approach for approximate solution of integer and mixed integer problems, see for example [8] and [7]. The Lagrangian relaxation of SPLPE is obtained by relaxing the demand constraints, using multipliers α_j. We obtain the following Lagrangian dual:

$$\text{(LD)} \quad v_L = \max \varphi(\alpha),$$

where, for fixed multipliers, $\alpha = \overline{\alpha}$, the Lagrangian relaxation takes the following form:

$$\text{(DS)} \quad \begin{cases} \varphi(\overline{\alpha}) & = \min \displaystyle\sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{w_j} \overline{c}_{ijk} x_{ijk} \\ & \quad + \displaystyle\sum_{i=1}^{m} \gamma a_i z_i \\ & \quad + \displaystyle\sum_{j=1}^{n} \overline{\alpha}_j \left(\sum_{i=1}^{m} \sum_{k=1}^{w_j} x_{ijk} - w_j \right) \\ \text{s.t.} & x_{ijk} - z_i \leq 0, \quad \forall i, j, k, \\ & x_{ijk} \in \{0, 1\}, \quad \forall i, j, k, \\ & z_i \in \{0, 1\}, \quad \forall i. \end{cases}$$

(DS) separates into m problems, one for each facility, containing one binary variable and a number of continuous variables, and is trivially solvable. It has the 'integrality property', i. e. the y-solution will obtain integral values even if the integrality constraints are removed. This property implies that the optimal value of LD is the same as that of the LP relaxation.

In order to solve the Lagrangian dual, we use the well known technique subgradient optimization, see [25,26] and [9]. We use a subgradient, ξ, of the dual function $\varphi(\alpha)$ to find a search direction for updating the multipliers, α:

$$\xi_j = \sum_{i=1}^{m} \sum_{k=1}^{w_j} \overline{x}_{ijk} - w_j, \quad \forall j,$$

where $\overline{x}$ denotes the optimal solution of (DS) at $\overline{\alpha}$. Letting $\overline{\alpha}^{(l)}$ denote the multiplier values in iteration l, we obtain the multipliers in the next iteration by setting

$$\overline{\alpha}^{(l+1)} = \overline{\alpha}^{(l)} + t^{(l)} \xi^{(l)},$$

where $t^{(l)}$ and $\xi^{(l)}$ are the stepsize and the search direction. Several ways of choosing the stepsize, $t^{(l)}$, have been suggested. Here we use the one that is suggested by [26]:

$$t^{(l)} = \lambda_l \frac{\widetilde{v} - \varphi(\overline{\alpha}^{(l)})}{\left\| \xi^{(l)} \right\|^2},$$

where $\widetilde{v}$ is an upper bound of v_L and λ_l should be assigned a value in the interval $(\varepsilon_1, 2 - \varepsilon_1)$, where $\varepsilon_1 > 0$, in order to ensure convergence.

Termination of the subgradient search procedure occurs when $\|d^{(l)}\| < \epsilon$, $t^{(l)} < \epsilon$, $l > M$, $\overline{v} - \varphi(\overline{\alpha}^{(l)}) \le \epsilon$ or $\overline{v} - \underline{v} < 1$. The last criterion indicates optimality, since all feasible solutions are integral, i. e. v^* is integral.

Benders Decomposition

We have noted that if all z-variables were fixed, the solution would not be changed if the constraints $x_{ijk} \in \{0,1\}$ were replaced by $0 \le x_{ijk} \le 1$. Therefore one might regard SPLPE as a mixed integer programming problem. This opens up the possibility of solving the problem with Benders decomposition, [1]. Below we give a short description of how the method can be applied to SPLPE, as done in [16].

In the Benders subproblem, (PS), we fix z to $\overline{z}$, which makes the subproblem separable into several trivial knapsack problems:

$$\text{(PS)} \quad h(\overline{z}) = \sum_{j=1}^{n} h_j(\overline{z}) + \sum_{i=1}^{m} \gamma a_i \overline{z}_i,$$

where, $\forall$ j,

$$\begin{cases} h_j(\overline{z}) & = \min \sum_{i=1}^{m} \sum_{k=1}^{w_j} \overline{c}_{ijk} x_{ijk} \\ \text{s.t.} & \sum_{i=1}^{m} \sum_{k=1}^{w_j} x_{ijk} = w_j, \\ & x_{ijk} \le \overline{z}_i, \quad \forall i, \\ & x_{ijk} \in \{0,1\}, \quad \forall i, k. \end{cases}$$

(PS) is feasible if and only if $\sum_i \overline{z}_i \ge 1$. The dual solution (α, β) is also easy to calculate.

The Benders master problem is given below.

$$\text{(PM)} \begin{cases} v_{\text{PM}} & = \min \sum_{j=1}^{n} q_j + \sum_{i=1}^{m} \gamma a_i z_i \\ \text{s.t.} & q_j \ge w_j \alpha_j^{(l)} - \sum_{i=1}^{m} \sum_{k=1}^{w_j} \beta_{ijk}^{(l)} z_i, \\ & \quad \forall l, j, \\ & \sum_{i=1}^{m} z_i \ge 1, \\ & z_i \in \{0,1\}, \quad \forall i. \end{cases}$$

The Benders decomposition method is to iterate between the master problem, (PM), and the subproblem, (PS). (PM) yields a lower bound on v^*, and $\overline{z}$ to be used in (PS). (PM) yields an upper bound on v^* (for integral $\overline{z}$) and a new dual solution, $(\alpha_j^{(l)}, \beta_{ijk}^{(l)})$, which is used to form a new cut for the master problem. The method has exact finite convergence.

The proportion of z-variables is much smaller in SPLPE than in SPLP, which is promising for the Benders decomposition approach. However, as shown computationally in [16], (PM) often is very difficult to solve. A suggested modification, [24], is to use the *LP relaxation* of (PM), by replacing $z_i \in \{0, 1\}$ with $0 \le z_i \le 1$, in initial iterations (for example until the LP-bounds are within 1% of each other). A good set of Benders cuts is thus generated before the integer master problem is solved. It is possible since any dual feasible solution of (PS) yields a valid Benders cut, and $\overline{z}$ only appears in the dual objective function.

If $\overline{z}$ is not integer, (PS) might not yield integer x-solutions, but is still easily solvable. The bounds obtained from the master problem and subproblem are not valid for the integer problem, but for the LP relaxation of SPLPE.

Mean Value Cross Decomposition

An alternate way of solving the LP relaxation of the problem is to use the method mean value cross decomposition, [10,11,12]. This method is a modification of the subproblem phase of ordinary cross decomposition, [28], but also a generalization of the Kornai–Liptak method, [21], and a generalization of the Brown–Robinson methods for polyhedral games, [2,27].

The method uses the Lagrangian relaxation and the subproblem of Benders decomposition, both described in previous sections, but no master problems. The input to one of the problems consists of the mean value of all the previous solutions of the other subproblem. The method has asymptotic convergence.

Comparisons and the Role of Γ

The parameter γ reflects the relation between the transportation costs and the effects of the spatial interaction in the objective function, and its value should be chosen specifically for each real life situation.

For very small values of γ, the optimal solution is $z_i = 1$, $\forall\ i$, while for large values of γ, the optimal z-solution can be obtained by DUALOC. In these cases the problem is then completely solved by simply solving the primal subproblem, (PS), once. In computational tests in [16,17,18] and [13], this occurs when γ is smaller than 0.0001 or larger than 0.1, while for $\gamma = 0.01$ the differences to the solutions mentioned above are the largest.

The conclusions of the computational tests in [16,17,18] and [13] are the following. Ordinary Benders decomposition seems to be more efficient than direct solution with a general integer programming code. However, direct solution with a standard IP-code can only solve small problems, due to memory requirements, and the ordinary Benders decomposition method also fails for many of the problems. The integer master problem is simply too hard.

The modified Benders decomposition method (starting with the LP relaxation of the master problem) eliminates the weaknesses of the Benders approach, and is a very efficient method.

The approximate methods mean value cross decomposition and Lagrangian relaxation with dual subgradient optimization are much quicker than ordinary Benders, but not better than modified Benders decomposition. For some large problems, these methods give large gaps between the upper and lower bounds.

The dual ascent method is also quite quick, but leaves gaps between the upper and lower bounds of varying size. In [18] it is noted that the dual ascent method and the Lagrangian method complement each other in an interesting way.

The best methods seems to be the modified Benders decomposition method and the dual ascent method with branch and bound. These methods are capable of solving quite large problems (up to almost 3,000,000 variables) optimally.

Finally we wish to point out that none of these methods explicitly store the whole x-matrix, and that this is what enables the solving of large problems.

Conclusion

We have described the simple plant location problem with spatial interaction, applied an exact linearization to the problem, and described a couple of solution methods for the resulting large integer programming problem. Although the model has a large number of variables, the methods are able to solve it quite efficiently. The problem is very well suited for the approaches of Benders decomposition, Lagrangian relaxation, and dual ascent. These methods actually manage to solve the problem without storing all of the variables, and especially the dual ascent method uses relatively small amounts of computer memory.

We conclude that the model is solvable and useful in practice.

See also

- ▶ Combinatorial Optimization Algorithms in Resource Allocation Problems
- ▶ Competitive Facility Location
- ▶ Facility Location with Externalities
- ▶ Facility Location with Staircase Costs
- ▶ Global Optimization in Weber's Problem with Attraction and Repulsion
- ▶ MINLP: Application in Facility Location-allocation
- ▶ Multifacility and Restricted Location Problems
- ▶ Network Location: Covering Problems
- ▶ Optimizing Facility Location with Rectilinear Distances

- ► Production-distribution System Design Problem
- ► Resource Allocation for Epidemic Control
- ► Single Facility Location: Circle Covering Problem
- ► Single Facility Location: Multi-objective Euclidean Distance Location
- ► Single Facility Location: Multi-objective Rectilinear Distance Location
- ► Stochastic Transportation and Location Problems
- ► Voronoi Diagrams in Facility Location
- ► Warehouse Location Problem

References

1. Benders JF (1962) Partitioning procedures for solving mixed-variables programming problems. Numerische Math 4:238–252
2. Brown GW (1949) Some notes on computation of games solutions. RAND Report P-78
3. Coelho JD (1980) A locational surplus maximization approach to public facility location. Nota Centro de Estasica e Aplic Dept Mat Aplic Fac Ciences de Lisboa, Portugal, no. 15
4. Coelho JD, Wilson AG (1976) The optimum location and size of shopping centres. Regional Studies 4:413–421
5. Erlenkotter D (1978) A dual-based procedure for uncapacitated facility location. Oper Res 26:992–1009
6. Erlenkotter D, Leonardi G (1985) Facility location with spatially interactive travel behaviour. Sistemi Urbani 1:29–41
7. Fisher ML (1981) The Lagrangean relaxation method for solving integer programming problems. Managem Sci 27(1):1–18
8. Geoffrion AM (1974) Lagrangean relaxation for integer programming. Math Program Stud 2:82–114
9. Held M, Wolfe P, Crowder HP (1974) Validation of subgradient optimization. Math Program 6:62–88
10. Holmberg K (1992) Linear mean value cross decomposition: A generalization of the Kornai–Liptak method. Europ J Oper Res 62:55–73
11. Holmberg K (1994) A convergence proof for linear mean value cross decomposition. Z Oper Res 39(2):157–186
12. Holmberg K (1997) Mean value cross decomposition applied to integer programming problems. Europ J Oper Res 97:124–138
13. Holmberg K (1999) Exact solution methods for uncapacitated location problems with convex transportation costs. Europ J Oper Res 114:127–140
14. Holmberg K, Jörnsten K (1989) Exact methods for gravity trip distribution models. Environm Plan A 21:81–97
15. Holmberg K, Jörnsten K (1990) An exact formulation of the simple plant location problem with spatial interaction. In: Orban-Ferauge Fr, Rasson JP (eds) Proc Meeting V of the EURO Working Group on Locational Analysis, Fac Univ Notre-Dame de-la-Paix, Namur
16. Holmberg K, Jörnsten K (1995) Decomposition methods for the exact formulation of the simple plant location problem with spatial interaction. Res Report, Dept Math, Linköping Inst Techn, LiTH-MAT-R-1995–13
17. Holmberg K, Jörnsten K (1996) A dual ascent procedure for the exact formulation of the simple plant problem with spatial interaction. Optim 36:139–152
18. Holmberg K, Jörnsten K (1996) Dual search procedures for the exact formulation of the simple plant location problem with spatial interaction. Location Sci 4:83–100
19. Jacobsen SK (1986) Plant location by entropy maximization. Res Report IMSOR, DTH, Lyngby, Denmark 10
20. Jacobsen SK (1988) On the solution of an entropy maximizing location model by submodularity. Res Report, IMSOR, DTH, Lyngby, Denmark 6
21. Kornai J, Liptak Th (1965) Two-level planning. Econometrica 33:141–169
22. Leonardi G (1978) Optimum facility location by acessibility maximizing. Environm Plan A 11:1287–1305
23. Leonardi G (1983) The use of random-utility theory in building location-allocation models. In: Thisse JF, Zoller HG (eds) Location Analysis of Public Facilities. North-Holland, Amsterdam, pp 357–383
24. McDaniel D, Devine M (1977) A modified Benders partitioning algorithm for mixed integer programming. Managem Sci 24:312–319
25. Poljak BT (1967) A general method of solving extremum problems. Soviet Math Dokl 8:593–397
26. Poljak BT (1969) Minimization of unsmooth functionals. USSR Comput Math Math Phys 9:14–29
27. Robinson J (1951) An iterative method of solving a game. Ann of MATH 54:296–301
28. Roy TJ Van (1983) Cross decomposition for mixed integer programming. Math Program 25:46–63
29. Wilson AG (1967) A statistical theory of spatial distribution models. Transport Res 1:253–269

Facility Location with Staircase Costs

KAJ HOLMBERG
Department Math., Linköping Institute Technol.,
Linköping, Sweden

MSC2000: 90B80, 90C11

Article Outline

Keywords
Mathematical Model
Solution Methods
Primal Heuristics
Linearization

Convex Piecewise Linearization
Benders Decomposition
Lagrangian Relaxation
and Subgradient Optimization
Computational Results
Conclusion
See also
References

Keywords

Facility location; Mixed integer programming; Heuristics

Location of facilities, plants, or other units for production or distribution, is an important problem in many different situations. The same type of problem can occur when one is installing equipment in for example telecommunication networks, or when installing machines in a factory.

The common circumstances in these situations are the following. A number of units, 'facilities', producing a certain service, may be located at certain possible points. The service commodity is then to be sent from the facilities to certain 'customer' points, which have a certain demand for the service. The main complication is that the costs for production of the service is not linear, instead there is a fixed cost for placing a facility at a certain location. In addition there may be linear costs for producing and shipping the commodity to the customers.

In the literature, see for example [3,4,6,8,13] and [1], one can find the traditional capacitated plant location model, where the total cost for satisfying demand consists of two parts, namely linear transportation costs and fixed costs for opening/installing the facilities. In this model there is one fixed cost for each facility. (Other variants can be found in ► Stochastic transportation and location problems and ► Facility location problems with spatial interaction.)

However, in practice, there is often a need for considering several different possible sizes of each facility. This leads to a *facility location problem* with *staircase* shaped costs. This approach will not only allow different sizes, but also different production costs at different levels of production at a facility.

For example, in telecommunications there are almost always several different sizes for the fibers, cables, switches, controllers and other connections that must be dimensioned when installing a new network. In such problems staircase shaped costs will occur at several different levels, both for the activities at nodes as well as along links. One situation where the specific location model discussed here is quite appropriate is the installation of video servers for a video-on-demand service on a telecommunication network.

Mathematical Model

We define a *staircase cost function* as a finite piecewise linear nondecreasing function with a finite set of discontinuities, each corresponding to a certain size of a facility. Let m be the number of possible location sites, n the number of customers and q_i the number of possible sizes at location site i. Furthermore, D_j is the demand of customer j, p_{ik} is the unit cost of production at a facility at location site i and size k, S_{ik} is the capacity of a facility of size k at location site i, f_{ik} is the fixed cost for a facility of size k at location site i, and c_{ij} is the cost for sending one unit from location site i to customer j.

The following variables are used: t_{ik} is the production within level k at facility i (where level k of the staircase corresponds to an operating facility of size k), x_{ij} is the amount shipped from location i to customer j, and y_{ik} is set to 1 if the facility at site i is of size k or larger and 0 otherwise.

The capacities and costs for increasing the size of a facility are $\Delta S_{ik} = S_{ik} - S_{ik-1}$ and $\Delta f_{ik} = f_{ik} - f_{ik-1} - (p_{ik} - p_{ik-1}) S_{ik-1}$, where $S_{i0} = 0$ and $f_{i0}, = 0$, see Fig. 1. Note that $0 \leq t_{ik} \leq \Delta S_{ik}$, $\forall \mathrm{i}, \mathrm{k}$, and if the total production at facility i requires more than size k, then $t_{ik} = \Delta S_{ik}$.

$$v^* = \min \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} + \sum_{i=1}^{m} \sum_{k=1}^{q_i} (p_{ik} t_{ik} + \Delta f_{ik} y_{ik}),$$

such that

$$\sum_{i=1}^{m} x_{ij} = D_j, \quad \forall j, \tag{1}$$

$$\sum_{j=1}^{n} x_{ij} = \sum_{k=1}^{q_i} t_{ik}, \quad \forall i, \tag{2}$$

$$t_{ik} \le \Delta S_{ik} y_{ik}, \quad \forall i, k, \tag{3}$$

$$t_{ik-1} \ge \Delta S_{ik-1} y_{ik}, \quad \forall i, k > 1, \tag{4}$$

$$x_{ij} \ge 0, \quad \forall i, k, \tag{5}$$

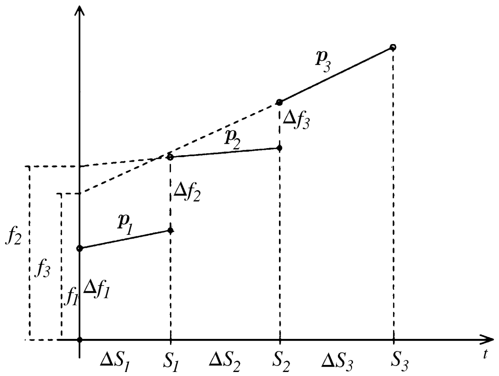
$$y_{ik} = 0 \text{ or } 1, \quad \forall i, k. \tag{6}$$

It is natural to assume that $p_{ik} \ge 0$, $\Delta f_{ik} \ge 0$, $\Delta S_{ik} > 0$ for all i, k and $D_j > 0$ for all j. The constraints (1) ensure that all the demand must be met for each customer, while (2) ensure that, for each location, the amount shipped also is produced. Constraint sets (3) and (4) ensure that the level of production corresponds to the correct level on the staircase cost function for each facility. One might note that from constraints (3) and (4) follows that $y_{ik+1} \le y_{ik}$.

This is a linear mixed integer programming problem with $mn + \sum_{i=1}^{m} q_i$ continuous variables and $\sum_{i=1}^{m} q_i$ integer variables. The proportion of integer variables is higher than in the ordinary facility location problem. Because of this, and because of the structure of the problem, solving the problem with a general code for mixed integer programming problems is probably not very efficient for large (real life) instances.

One aspect of the structure of the problem is that if y is kept fixed (i. e. the sizes of the facilities are given), the remaining problem is simply a standard network flow problem, and hence x and t will attain integer values.

Another important aspect of the structure of the problem is the potential separability. There are several possibilities of making the model separable by relaxing different sets of constraints.



Facility Location with Staircase Costs, Figure 1

It is also possible to use a problem formulation with f and S instead of Δf and ΔS. This yields constraints of SOS1-type (one must ensure that only one of the possible sizes is used at a facility), and a somewhat smaller problem (less constraints). The LP relaxation is quicker to solve and the optimal objective function value is the same as that of the model above (i. e. the duality gaps of the two formulations are the same). However, solving the model with general mixed integer codes, the alternate model seems to produce larger branch and bound trees. Concerning the methods discussed below, the two models in most cases behave in identical manners.

Solution Methods

Methods for models with staircase cost functions or for models capable of modeling such functions can be found in for example [2,11,14,15] and [12]. We will below describe some possibilities.

If the exact solution is to be found (and verified), the only reasonable way seems to be to resort to branch and bound, in some sense. This matter in general is extensively discussed in the literature, and although there might be some considerations for the staircase cost case that differ from the single fixed cost case, when it comes to branching and search strategies, we will not dwell on it here.

Assuming a standard branch and bound framework, the main question is how to solve the subproblems, i. e. how to get the bounds, especially the lower bounds. This will be discussed more below.

However, an alternative is to move the branch and bound procedure into a Benders master problem, i. e. use a Benders decomposition framework in order to obtain the exact solution. This will also be briefly described below.

We will start with procedures for obtaining upper and lower bounds on the optimal objective function value. The upper bounds correspond to feasible solutions obtained, while the lower bounds are used to get estimates of the quality of the feasible solutions. If all cost coefficients are integral, we note that a lower bound that is within one unit from the upper bound indicates that the upper bound is optimal.

Primal Heuristics

There is a well-known ADD heuristic for the capacitated plant location problem, [13], which can be modified to suit the staircase cost facility location problem, see [12]. This heuristic can be improved by combining it with certain priority rules, [6].

If for each plant it is decided to which level of production it can be used (i. e. the y-variables are fixed), the resulting problem is an unbalanced transportation problem. Let L_i denote the level (size) of plant i, and initiate the heuristic by setting $L_i = 0, \forall i$. Let $I = \{i\colon L_i < q_i\}$. The ADD heuristic consists of the repeated use of the following step: Increase the size (set $L_i = L_i + 1$) of the location site $i \in I$ that provides the largest reduction of the total cost. Terminate the procedure when no more reduction is possible.

In order to avoid ADD increasing the level of production in the order of 'decreasing' capacity until a feasible solution is found, we apply a generalization of one of the priority rules discussed in [6]. These priority rules provides a better phase-1 solution than the ADD heuristic itself. Two examples of priority rules, PR1 and PR2, for choosing the location site $i \in I$ where the size is to be increased ($L_i = L_i + 1$), are given below. (They correspond to P1 and P3 in the notation of [6]).

PR1) Choose site $i \in I$ in the order of decreasing quotients $\Delta S_{i,L_i+1}/\Delta f_{i,L_i+1}$, until the location sites are able to serve the entire demand.

PR2) Choose site $i \in I$ in the order of increasing values of

$$\frac{1}{\lfloor n/3 \rfloor} \sum_{j=1}^{\lfloor n/3 \rfloor} \bar{c}_{ij} + \frac{\Delta f_{i,L_i+1}}{\Delta S_{i,L_i+1}},$$

until the location sites are able to serve the entire demand. ($\bar{c}$ is c sorted according to increasing values.)

In [13] the ADD heuristic is outperformed by the heuristic DROP but [6] show that ADD with priority rules produce solutions with equally good objective function values as DROP, in less computational time.

Linearization

A widely used way of obtaining a lower bound is direct LP relaxation. The integer requirements (6) are replaced with the constraints $0 \leq y_{ik} \leq 1, \forall i, k$. We also include the redundant constraints $t_{ik} \leq \Delta S_{ik}, \forall i, k$, and possibly $y_{ik} \leq y_{ik-1}$ for all $i, k > 1$. The optimal objective function value of the LP relaxation is denoted by v_{LP}, and $v_{\text{LP}} \leq v^*$. The duality gap, the difference between v^* and v_{LP}, is in most cases larger than zero. The LP-problem is large, but sparse, and can be solved with a standard LP-code.

Convex Piecewise Linearization

Since the binary variables y_{ik} are only included to give the correct cost for the production, they can be eliminated if we use an approximation of the costs. If the staircase cost function is underestimated with a piecewise linear and convex function, we get a problem, much easier to solve, which gives a lower bound on v^*, denoted by v_{CPL}, see [14] and [11]. For explicit expressions of how to construct the convex piecewise linearization see [11].

The resulting problem is a linear minimal cost network flow problem with parallel arcs, which is quite easily solvable by a standard network code. The x- and t-part of the solution is feasible in the original problem, so we can generate an upper bound by evaluating this solution in the correct cost function, which is done by finding the correct values of y.

In [10] it is proved that the convex piecewise linearization and the LP relaxation are equivalent, in the sense that $v_{\text{CPL}} = v_{\text{LP}}$ and an x-solution that is optimal in one of the problems is also optimal in the other problem. Utilizing the network structure, we thus get a quicker way of solving the LP relaxation.

Benders Decomposition

In [11] a Benders decomposition approach is used, and combined with the convex piecewise linearization described above.

The Benders subproblem is simply obtained by fixing the integer variables, i. e. fixing the sizes of the facilities. The resulting problem is minimal cost network flow problem, similar to a transportation problem, but with certain intervals (given by the facility sizes) for the supplies.

However, the Benders master problem obtained by a standard application of the Benders approach, is much too hard to solve. The number of integer variables is much larger than in an ordinary location problem with the same numbers of facilities and customers. One way around this is to combine the Benders approach with the convex piecewise linearization.

An *improved piecewise linearization* is obtained by branching at certain production levels. A staircase cost function is divided into two parts by the branching, and a binary variable is introduced, indicating which of the parts that is to be used. In each of the two parts, convex piecewise linearization is used. In this manner, one could design a branch and bound method for solving the problem, similar to what is described in [14].

Considering the model after a number of branchings, we have an approximation (a relaxation) of the original problem, with a much smaller number of integer variables. On this problem we then apply Benders decomposition.

In principle one could let each subproblem in the branch and bound method be solved exactly with Benders decomposition, thereby obtaining basically a branch and bound method, which employs Benders decomposition to solve the branch and bound subproblems. This is however very inefficient.

The other extreme is standard application of Benders decomposition to the original problem, in which case the Benders approach employs branch and bound to solve the Benders subproblems. This is also quite inefficient in practice.

A more efficient method is to combine the two approaches, Benders decomposition and branch and bound on a more equal level. This can be done the following way.

1) Solve the initial convex piecewise linearization (with a network code).
2) Do one or more branchings, where the error of the approximation at the obtained solution is largest.
3) Solve the obtained problem with Benders decomposition (to a certain accuracy).
4) Repeat 2) and 3), until optimality.

There are two very important comments to the above algorithm.

A) When one returns to step 3) after having done branchings, one can recalculate and reuse all the Benders cuts obtained before the branchings. (This is described in detail in [11].)
B) The stopping criterion for the Benders method, i. e. the required accuracy in step 3), is a very important control parameter. One should in initial iterations require a low accuracy, and gradually, as the method approaches the optimal solution, require higher and higher accuracies.

The effect of combining comments A) and B) is that one should only do a few Benders iteration in each main iteration, since the number of Benders cuts will automatically increase, as the old cuts are recalculated and kept.

The main conclusion of the computational tests done in [11] is that only a small part of all the integer variables (in average 4%) need to be included by the improving piecewise linearization technique, when solving a problem to reasonable accuracy. In other words, only a small subset of the possible sizes need to be investigated.

Lagrangian Relaxation and Subgradient Optimization

Now we will describe a Lagrangian heuristic, found in [12], in more detail. Lagrangian relaxation and subgradient optimization are used to obtain a near-optimal dual solution, and act as a base for an efficient primal heuristic. Based on the solution of the Lagrangian relaxation one can construct a transportation problem which yields primal feasible solutions, and can be used during the subgradient process.

An important aspect of the Lagrangian approach is that a method yielding good feasible *primal solutions* can be based on *dual techniques*.

Lagrangian relaxation, [7], in combination with subgradient optimization, [9] is a commonly used technique for generating lower bounds on the optimal objective function value of mixed integer programming problems. Here we apply Lagrangian relaxation to constraint set (2), and denote the Lagrangian multipliers by u_i. For fixed values of u, the subproblem separates into several smaller problems:

$$\begin{cases} \theta_{1j}(\overline{u}) & = \min \sum_{i=1}^{m} (c_{ij} + \overline{u}_i) x_{ij} \\ \text{s.t.} & \sum_{i=1}^{m} x_{ij} = D_j , \\ & x_{ij} \geq 0 , \end{cases}$$

$$\begin{cases} \theta_{2i}(\overline{u}) & = \min \sum_{k=1}^{q_i} ((p_{ik} - \overline{u}_i)t_{ik} + \Delta f_{ik} y_{ik}) \\ \text{s.t.} & t_{ik} \leq \Delta S_{ik} y_{ik}, \quad \forall k, \\ & t_{ik-1} \geq \Delta S_{ik-1} y_{ik}, \quad \forall k > 1, \\ & t_{ik} \geq 0, \\ & y_{ik} = 0 \text{ or } 1 \end{cases}$$

The first set of subproblems consists of n continuous knapsack problems, which are trivially solvable. The second set of subproblems consists of m one-dimensional staircase cost problems. The solution can be found by calculating the minimizer $\overline{k}_i$ for each i, as follows:

$$\theta_{2i}(\overline{u}_i) = \min_{k_i = 0, \ldots, q_i} \sum_{k=1}^{k_i} ((p_{ik} - \overline{u}_i)t_{ik} + \Delta f_{ik} y_{ik}).$$

The resulting solution is

$$y'_{ik} = \begin{cases} 1, & \forall k \leq \overline{k}_i, \\ 0, & \forall k > \overline{k}_i, \end{cases}$$

$$t'_{ik} = \begin{cases} \Delta S_{ik}, & \forall k \leq \overline{k}_i, \\ 0, & \forall k > \overline{k}_i. \end{cases}$$

Note that the subproblem has the integrality property, [7], so max $\theta(u) = v_{\mathrm{LP}}$.

The Lagrangian dual,

$$\max \theta(u) = \sum_{j=1}^{n} \theta_{1j}(u) + \sum_{i=1}^{m} \theta_{2i}(u)$$

can be solved by standard subgradient optimization, [9], in order to get the best lower bound. One can use enhancements such as modified directions, [5], $d^r = \xi^r + \alpha\, d^{r-1}$, where ξ^r is the subgradient generated in iteration r and d^r is the direction used in iteration r.

A steplength shortening is obtained by setting $\lambda = \lambda \backslash 2$ when there has not been any improvement of $\underline{v}$ for $\overline{N}_1$ consecutive iterations. When there has not been any improvement of $\underline{v}$ for $\overline{N}_2$ iterations, the subgradient optimization procedure is terminated. The subgradient is given by $\xi_i^r = \sum_{j=1}^{n} x_{ij}' - \sum_{k=1}^{q_i} t_{ik}'$ for all i, where x_{ij}' and t_{ik}' are the optimal solutions to the subproblems. Reasonable choices for the parameters are $\overline{N}_1 = 6$, $\overline{N}_2 = 25$, and $\alpha = 0.7$.

One can use a heuristic based on the solution of the Lagrangian relaxation to try to get a feasible solution. The obtained values of y_{ik}' are used to calculate the supply at each location and a transportation problem is solved. The solution to the transportation problem is feasible in the original problem if constraint sets (3) and (4) are satisfied, which easily can be achieved. The values of the flow variables x_{ij} are taken directly from the solution to the transportation problem. The total production t_i is then calculated as $t_i = \sum_{j=1}^{n} x_{ij}$. One can then easily find t_{ik} as the part of t_i that lies within level k, and the y_{ik} solution is simply $y_{ik} = 1$ if $t_{ik} > 0$ and 0 if not. Finally all unnecessary production capacity at each location i is removed.

The complete Lagrangian heuristic, LH, also includes the following. The convex piecewise linearization, CPL, is solved with an efficient network code. The Lagrangian multipliers are initiated with a convex combination of the appropriate node prices obtained by solving CPL and $\min_j c_{ij}$, with the largest weight on the former. The primal procedure to generate feasible solutions is used every third iteration in the subgradient procedure.

Note that CPL yields $v_{\mathrm{CPL}} = v_{\mathrm{LP}}$, so the subgradient procedure cannot improve the lower bound, which is quite unusual in methods of this kind. The motivation behind using the subgradient procedure is not to get lower bounds, but to get primal solutions (upper bounds).

Computational Results

In [12] the heuristic procedures are tested by solving a number of randomly generated test problems, with up to 50 locations, 100 destinations and 20 sizes of each location (yielding 6000 continuous variables and 1000 integer variables). The conclusions of the tests are the following.

A standard mixed integer programming code (in this case LAMPS) needs extremely long solution times for finding the exact optimum. The ADD heuristics produce solutions with relative errors in the range of 1%–20% (in average 11%), but also requires quite long solution times (although not as long as the MIP-code).

The convex piecewise linearization CPL, combined with exact integer evaluation of the solutions obtained, yields solutions that all are better than those obtained by the ADD heuristics, with relative errors between 0.8% and 10% (in average 4%), in a much shorter time (in average 1000 times quicker than the ADD heuristics). So CPL dominates the ADD heuristics completely, both with respect to solution time and solution quality.

The Lagrangian heuristic, LH, produces solutions with relative errors between 0.4% and 3.2% (in average 1.5%), with solution times in average 20 times shorter than the ADD heuristics, but of course significantly longer than CPL.

Comparison to other tests is difficult, since other computers and codes are used. The Benders approach in [11] seems to be slower than the Lagrangian approach. However, on modern computers and with modern MIP-codes, its performance may well improve.

Conclusion

The capacitated facility location problem with staircase costs has many important applications. Computational results indicate that it is possible to find near-optimal solutions to such problems of reasonable size in a reasonable time, i. e. that this better model can be used instead of, for example, the ordinary capacitated facility location problem in appropriate situations.

See also

- ► Combinatorial Optimization Algorithms in Resource Allocation Problems
- ► Competitive Facility Location
- ► Facility Location with Externalities
- ► Facility Location Problems with Spatial Interaction
- ► Global Optimization in Weber's Problem with Attraction and Repulsion
- ► MINLP: Application in Facility Location-allocation
- ► Multifacility and Restricted Location Problems
- ► Network Location: Covering Problems
- ► Optimizing Facility Location with Rectilinear Distances
- ► Production-distribution System Design Problem
- ► Resource Allocation for Epidemic Control
- ► Single Facility Location: Circle Covering Problem
- ► Single Facility Location: Multi-objective Euclidean Distance Location
- ► Single Facility Location: Multi-objective Rectilinear Distance Location
- ► Stochastic Transportation and Location Problems
- ► Voronoi Diagrams in Facility Location
- ► Warehouse Location Problem

References

1. Beasley JE (1993) Lagrangean heuristics for location problems. Europ J Oper Res 65:383–399, Testproblems available at http://mscmga.ms.ic.ac.uk
2. Bornstein CT, Rust R (1988) Minimizing a sum of staircase functions under linear constraints. Optim 19:181–190
3. Christofides N, Beasley JE (1983) Extensions to a Lagrangean relaxation approach for the capacitated warehouse location problem. Europ J Oper Res 12:19–28
4. Cornuejols G, Sridharan R, Thizy JM (1991) A comparison of heuristics and relaxations for the capacitated plant location problem. Europ J Oper Res 50:280–297
5. Crowder H (1976) Computational improvements for subgradient optimization. Symp Math, vol XIX. Acad. Press, pp 357–372
6. Domschke W, Drexl A (1985) ADD-heuristics' starting procedures for capacitated plant location models. Europ J Oper Res 21:47–53
7. Geoffrion AM (1974) Lagrangean relaxation for integer programming. Math Program Stud 2:82–114
8. Geoffrion A, McBride R (1978) Lagrangean relaxation applied to capacitated facility location problems. AIIE Trans 10:40–47
9. Held M, Wolfe P, Crowder HP (1974) Validation of subgradient optimization. Math Program 6:62–88
10. Holmberg K (1991) Linearizations of the staircase cost facility location problem. Res Report Dept Math Linköping Inst Techn, no. LiTH-MAT-R-1991-19
11. Holmberg K (1994) Solving the staircase cost facility location problem with decomposition and piecewise linearization. Europ J Oper Res 75:41–61
12. Holmberg K, Ling J (1997) A Lagrangean heuristic for the facility location problem with staircase costs. Europ J Oper Res 97:63–74
13. Jacobsen SK (1983) Heuristics for the capacitated plant location model. Europ J Oper Res 12:253–261
14. Rech P, Barton LG (1970) A non-convex transportation algorithm. In: Beale EM (ed) Applications of Mathematical Programming Techniques, pp 250–260
15. Sridharan R (1991) A Lagrangean heuristic for the capacitated plant location problem with side constraints. J Oper Res Soc 42:579–585

Farkas Lemma

Fl

Kees Roos
Department ITS/TWI/SSOR,
Delft University Technol., Delft, The Netherlands

MSC2000: 15A39, 90C05

Article Outline

Keywords
See also
References

Keywords

Inequality systems; Certificate; Theorem of the alternative; Skew-symmetric matrix; Orthogonal matrix

Farkas' lemma is the most well-known *theorem of the alternative* or *transposition theorem* (cf. ► Linear optimization: Theorems of the alternative). Given an $m \times n$ matrix A and a vector b (of dimension m) it states that either the set

$$S := \left\{y \colon y^\top A \geq 0, y^\top b < 0\right\}$$

or the set

$$T := \{x \colon \ Ax = b, \ x \geq 0\}$$

is empty but not both sets are empty. This result has a long history and it has had a tremendous impact on the development of the duality theory of linear and nonlinear optimization.

J. Farkas (1847–1930) was professor of Theoretical Physics at the Univ. of Kolozsvár in Hungary. His interest in the subject is explained in the first two sentences of his paper [5]:

> The natural and systematic treatment of analytic mechanics has to have as its background the inequality principle of virtual displacements first formulated by Fourier and later by Gauss. The possibility of such a treatment requires, however, some knowledge of homogeneous linear inequalities that may be said to have been entirely missing up to now.

J.B.J. Fourier [7] seems to have been the first who established that a mechanical system has a stable equilibrium state if and only if some homogeneous system of inequalities, like in the definition of the above set S, has no solution. This observation became known as the *mechanical principle of Fourier*. By Farkas'lemma this happens if and only if the set T is nonempty.

It is almost obvious that if the set T is not empty, then the set S will be empty and we have equilibrium. This follows easily by noting that the sets S and T cannot be both nonempty: if $y \in S$ and $x \in T$ then the contradiction

$$y^\top b = y^\top (Ax) = (y^\top A)x \geq 0$$

follows, because $y^\top \ A \geq 0$ and $x \geq 0$. This shows that the condition 'T is not empty' is certainly a sufficient condition for equilibrium. The hard part is to prove that this is also a necessary condition for equilibrium. The proof has a long history. First, the condition without proof for special cases was given by A. Cournot in 1827 and for the general case by M. Ostrogradsky in 1834. Farkas published his condition first in 1894 and 1895, but the proof contains a gap. A second attempt, in 1896, is also incomplete. The first complete proof was published in Hungarian, in 1898 [3], and in German in 1899 [4]. This proof is included in Farkas' best known paper [5]. For more details and references, see the historical overviews [9] and [10].

Nowadays (1998) many different proofs of Farkas' lemma are known. For quite recent proofs, see, e. g., [1,2,8]. An interesting derivation has been given by A.W. Tucker [11], based on a result that will be referred to as *Tucker's theorem*. (See ► Tucker homogeneous systems of linear relations.) The theorem states that for any *skew-symmetric matrix* K (i. e., $K = - K^\top$) there exists a vector x such that

$$Kx \geq 0, \quad x \geq 0, \quad x + Kx > 0.$$

By taking

$$K = \begin{pmatrix} 0 & 0 & A & -b \\ 0 & 0 & -A & b \\ -A^\top & A^\top & 0 & 0 \\ b^\top & -b^\top & 0 & 0 \end{pmatrix},$$

Tucker's theorem implies the existence of nonnegative vectors z_1, z_2 and x and a nonnegative scalar t such that

$$Ax - tb \geq 0, \tag{1}$$

$$- Ax + tb \geq 0, \tag{2}$$

$$\begin{aligned} -A^\top z_1 + A^\top z_2 &\geq 0, \\ b^\top z_1 - b^\top z_2 &\geq 0, \end{aligned} \tag{3}$$

and

$$\begin{aligned}
&z_1 + Ax - tb > 0,\\
&z_2 - Ax + tb > 0,\\
&x - A^\top z_1 + A^\top z_2 > 0,\\
&t + b^\top z_1 - b^\top z_2 > 0. \qquad (4)
\end{aligned}$$

If $t = 0$, then, putting $y = z_2 - z_1$, (3) and (4) yield a vector in the set S. If $t > 0$, since the above inequalities are all homogeneous, one may take $t = 1$ and then (1) and (2) give a vector in the set T. This shows that at least one of the two sets S and T is nonempty, proving the hard part of Farkas' lemma.

It is worth mentioning a *result* of C.G. Broyden [1] who showed that Tucker's theorem, and hence also Farkas' lemma, follows from a simple property of orthogonal matrices. The result states that for any orthogonal matrix Q (so $QQ^\top = Q^\top Q = I$) there exists a unique sign matrix D and a positive vector x such that $Qx = Dx$; a *sign matrix* is a diagonal matrix whose diagonal elements are equal to either plus one or minus one.

The key observation here is that if K is a skew-symmetric matrix, then

$$Q = (I + K)^{-1}(I - K)$$

is an orthogonal matrix, where I denotes the identity matrix; Q is known as the *Cayley transform* of K [6]. The proof of this fact is straightforward. First, for each vector x one has

$$x^\top (I + K)x = x^\top x,$$

whence $I + K$ is an invertible matrix. Furthermore, using $K^\top = -K$, one may write

$$\begin{aligned}
Q^\top Q &= (I + K)(I - K)^{-1}(I + K)^{-1}(I - K)\\
&= (I + K)(I - K^2)^{-1}(I - K).
\end{aligned}$$

Multiplying both sides from the left with $(I - K)$ one gets

$$\begin{aligned}
(I - K)QQ^\top &= (I - K^2)(I - K^2)^{-1}(I - K)\\
&= (I - K),
\end{aligned}$$

and multiplying both sides with $(I - K)^{-1}$ one finds $QQ^\top = I$, showing that Q is orthogonal indeed.

Therefore, by Broyden's theorem, there exists a sign matrix D and a positive vector z such that

$$(I + K)^{-1}(I - K)z = Dz.$$

This can be rewritten as

$$(I - K)z = (I + K)Dz,$$

whence

$$z - Kz = Dz + KDz,$$

or

$$z - Dz = K(z + Dz).$$

Defining $x = z + Dz$ one has $x \geq 0$, $Kx \geq 0$ and $x + Kx = 2z > 0$, proving Tucker's theorem.

See also

- ▶ Farkas Lemma: Generalizations
- ▶ Linear Optimization: Theorems of the Alternative
- ▶ Linear Programming
- ▶ Motzkin Transposition Theorem
- ▶ Theorems of the Alternative and Optimization
- ▶ Tucker Homogeneous Systems of Linear Relations

References

1. Broyden CG (1998) A simple algebraic proof of Farkas' lemma and related theorems. Optim Methods Softw 8:185–199
2. Dax A (1997) An elementary proof of Farkas' lemma. SIAM Rev 39:503–507
3. Farkas Gy (1898) A Fourier-féle mechanikai elv alkalmazásának algebrai alapja. Math Természettudományi Értesitö 16:361–364
4. Farkas J (1899) Die algebraischen Grundlage der Anwendungen des mechanischen Princips von Fourier. Math Naturwissenschaftl Bericht Ungarn 16:154–157
5. Farkas J (1902) Theorie der Einfachen Ungleichungen. J Reine Angew Math 124:1–27
6. Fekete A (1985) Real linear algebra. M. Dekker, New York
7. Fourier JBJ (1826) Solution d'une question particulière du calcul des inégalités. Nouveau Bull Sci Soc Philomath Paris, pp 99–100
8. Klafsky E, Terlaky T (1987) Remarks on the feasibility problem of oriented matroids. Ann Univ Sci Budapest R Eötvös 7:155–157
9. Prékopa A (1980) On the development of optimization theory. Amer Math Monthly 87:527–542

10. Schrijver A (1986) Theory of linear and integer programming. Wiley, New York
11. Tucker AW (1956) Dual systems of homogeneous linear relations. In: Kuhn HW, Tucker AW (eds) Linear Inequalities and Related Systems. Ann Math Stud., vol 38. Princeton Univ Press, Princeton, pp 3–18

Farkas Lemma: Generalizations

V. JEYAKUMAR
School of Math., University New South Wales,
Sydney, Australia

MSC2000: 46A20, 90C30, 52A01

Article Outline

Keywords
Infinite-Dimensional Optimization
Nonsmooth Optimization
Global Nonlinear Optimization
Nonconvex Optimization
Semidefinite Programming
See also
References

Keywords

Inequality systems; ∊-subdifferential; D.c. function; Global optimization; Convex inequality systems; Convex-like systems; Nonsmooth optimization; Semidefinite programming; Alternative theorem

The key to identifying optimal solutions of constrained nonlinear optimization problems is the Lagrange multiplier conditions. One of the main approaches to establishing such multiplier conditions for inequality constrained problems is based on the dual solvability characterizations of systems involving inequalities. J. Farkas [7] initially established such a dual characterization for linear inequalities which was used in [23] to derive necessary conditions for optimality for nonlinear programming problems. This dual characterization is popularly known as Farkas' lemma, which states that given any vectors $a_1, \dots, a_m$ and c in $\mathbf{R}^n$, the linear inequality $c^\top x \geq 0$ is a consequence of the linear system $a_i^\top x \geq 0$, $i = 1, \dots, m$, if and only if there exist multipliers $\lambda_i \geq 0$ such that $c = \sum_{i=1}^m \lambda_i a_i$. This result can also be expressed as a so-called *alternative theorem*: Exactly one of the following alternatives is true:

i) $\exists x \in \mathbf{R}^n,\ a_i^\top x \geq 0,\ c^\top x < 0$,
ii) $\exists \lambda_i \geq 0,\ c = \sum_{i=1}^m \lambda_i a_i$.

This lemma is the key result underpinning the linear programming duality and has played a central role in the development of nonlinear optimization theory. A large variety of proofs of the lemma can be found in the literature (see [5,25,26]). The proof [3,5] that relies on the separation theorems has led to various extensions. These extensions cover wide range of systems including systems involving infinite-dimensional linear inequalities, *convex inequalities* and matrix inequalities. Applications range from classical nonlinear programming to modern areas of optimization such as nonsmooth optimization and semidefinite programming. Let us now describe certain main generalizations of Farkas' lemma and their applications to problems in various areas of optimization.

Infinite-Dimensional Optimization

The Farkas lemma for a finite system of linear inequalities has been generalized to systems involving arbitrary convex cones and continuous linear mappings between spaces of arbitrary dimensions. In this case the lemma holds under a crucial closure condition. In symbolic terms, the main version of such extension to arbitrary dual pairs of vector spaces states that the following equivalence holds [6]:

$$\left[A(x) \in S \Rightarrow c(x) \geq 0\right] \ \Leftrightarrow \ c \in A^\top(S^*), \qquad (1)$$

provided the cone $A^\top(S^*)$ is closed in some appropriate topology. Here A is a continuous linear mapping between two Banach spaces, S is a closed convex cone having the dual cone S^* [5]. The closure condition holds when S is a polyhedral cone in some finite-dimensional space. For simple examples of nonpolyhedral convex cones in finite dimensions where the closure condition does not hold, see [1,5]. However, the following asymptotic version of Farkas' lemma holds without a closure condition:

$$\left[A(x) \in S \Rightarrow c(x) \geq 0\right] \ \Leftrightarrow \ c \in \mathrm{cl}(A^\top(S^*)), \qquad (2)$$

where $\mathrm{cl}(A^\top(S^*))$ is the closure of $A^\top(S^*)$ in the appropriate topology. These extensions resulted in the development of asymptotic and nonasymptotic first order necessary optimality conditions for infinite-dimensional smooth constrained optimization problems involving convex cones and duality theory for infinite-dimensional linear programming problems (see e. g. [12]). Smooth optimization refers to the optimization of a differentiable function. A nonasymptotic form of an extension of Farkas' lemma that is different from the one in (1) is given in [24] without the usual closure condition. For related results see [4]. An approach to the study of semi-infinite programming, which is based on generalized Farkas' lemma for infinite linear inequalities is given in [12].

Nonsmooth Optimization

The success of linear programming duality and the practical nature of the Lagrange multiplier conditions for smooth optimization have led to extensions of Farkas' lemma to systems involving nonlinear functions. Convex analysis allowed to obtain extensions in terms of *subdifferentials* replacing the linear systems by sublinear (convex and positively homogeneous) systems [8,31]. A simple form of such an extension states that the following statements are equivalent:

$$-g(x) \in S \Rightarrow f(x) \geq 0 \tag{3}$$

$$0 \in \operatorname{cl}\left[\partial f(0) + \bigcup_{\lambda \in S^*} \partial(\lambda g)(0)\right], \tag{4}$$

where the real valued function f is sublinear and lower semicontinuous, and the vector function g is sublinear with respect to the cone S and vg is lower semicontinuous for each $v \in S^*$. When f is continuous the statement (4) collapses to the condition

$$0 \in \partial f(0) + \operatorname{cl}\left[\bigcup_{\lambda \in S^*} \partial(\lambda g)(0)\right]. \tag{5}$$

This extension was used to obtain optimality conditions for convex optimization problems and *quasidifferentiable problems* in the sense of B.N. Pshenichnyi [27]. A review of results of Farkas type for systems involving sublinear functions is given in [13,14].

Difference of sublinear (DSL) functions which arise frequently in nonsmooth optimization provide useful approximations for many classes of nonconvex *nonsmooth functions*. This has led to the investigation of results of Farkas type for systems involving DSL functions.

A mapping g: $X \to Y$ is said to be *difference sublinear* (DSL) (with respect to S) if, for each $v \in S^*$, there are (weak $*$) compact convex sets, here denoted $\underline{\partial}(\mathrm{vg})(0)$ and $\overline{\partial}(vg)(0)$, such that, for each $x \in X$,

$$vg(x) = \max_{u \in \underline{\partial}(vg)(0)} u(x) - \max_{w \in \overline{\partial}(vg)(0)} w(x),$$

where X and Y are Banach spaces. If $Y = \mathbf{R}$ and $S = \mathbf{R}_+$ then this definition coincides with the usual notion of a difference sublinear real-valued function. Thus a mapping g is DSL if and only if vg is a DSL function for each $v \in S^*$. The sets $\underline{\partial}(\mathrm{vg})(0)$ and $\overline{\partial}(vg)(0)$ are the *subdifferential* and *superdifferential* of vg, respectively. For a DSL mapping g: $X \to Y$ we shall often require a *selection* from the class of sets $\left\{\overline{\partial}(vg)(0) \colon v \in S^*\right\}$. This is a set, denoted (w_v), in which we select a single element $\overline{\partial}(vg)(0)$ for each $v \in S^*$. An extension of the Farkas lemma for DSL systems states that the following statements are equivalent [10,20]:

i) $-g(x) \in S \Rightarrow f(x) \geq 0$;

ii) for each selection (w_v) with $w_v \in \overline{\partial}(vg)(0)$, $v \in S^*$, $\overline{\partial} f(0) \subseteq \underline{\partial} f(0) + B$,

where $B = \operatorname{cl}\operatorname{cone}\operatorname{co}\left[\bigcup_{v \in S^*}(\underline{\partial}(vg)(0) - w_v)\right]$. A unified approach to generalizing the Farkas lemma for sublinear systems which uses multivalued functions and convex process is given [2,17,18].

Global Nonlinear Optimization

Given that the optimality of a constrained *global optimization* problem can be viewed as the solvability of appropriate inequality systems, it is easy to see that an extension of Farkas' lemma again provides a mechanism for characterizing *global optimality* of a range of nonlinear optimization problems. The ϵ-*subdifferential* analysis here allowed to obtain a new version of the Farkas lemma replacing the linear inequality $c(x) \geq 0$ by a *reverse convex inequality* $h(x) \leq 0$, where h is a convex function with $h(0) = 0$. This extension for systems involving DSL functions states that the following conditions are equivalent.

i) $-g(x) \in S \Rightarrow h(x) \leq 0$;

ii) for each selection (w_v) with $w_v \in \overline{\partial}(vg)(0)$, $v \in S^*$ and for each $\epsilon \geq 0$,

$$\partial_\epsilon h(0) \subseteq \operatorname{cl}\operatorname{cone}\operatorname{co}\left[\bigcup_{v \in S^*} (\underline{\partial}(vg)(0) - w_v)\right].$$

Such an extension has led to the development of conditions which characterize optimal solutions of various classes of global optimization problems such as convex maximization problems and fractional programming problems (see [19,20]).

However, simple examples show that the asymptotic forms of the above results of Farkas type do not hold if we replace the DSL (or sublinear) system by a convex system. Ch.-W. Ha [15] established a version of the Farkas lemma for convex systems in terms of *epigraphs* of *conjugate functions*. A simple form of such a result [29] states that the following statements are equivalent:

i) $(\forall i \in I)\, g_i(x) \leq 0 \Rightarrow h(x) \leq 0$;
ii) $\operatorname{epi} h^* \subseteq \operatorname{cl}\operatorname{cone}\operatorname{co}\,[\,\cup_{i \in I} \operatorname{epi} g_i^*]$,
provided the system

$$i \in I, \quad g_i(x) \leq 0$$

has a solution. Here h and, for each $i \in I$, g_i are continuous convex functions, I is an arbitrary index set, and h^* and g_i^* are conjugate functions of h and g_i respectively. This result has also been employed to study infinite-dimensional nonsmooth nonconvex problems [30]. A basic general form of the Farkas lemma for convex system with application to *multiobjective convex optimization* problems is given in [11]. Extensions to systems involving the difference of convex functions are given in [21,29]. A more general result involving *H-convex functions* [29] with application to global nonlinear optimization is given in [29].

Nonconvex Optimization

The convexity requirement of the functions involved in the extended Farkas lemma above can be relaxed to obtain a form of Farkas' lemma for convex-like system. Let $F\colon X \times Y \to \mathbf{R}$ and let $f\colon X \to \mathbf{R}$, where X and Y are arbitrary nonempty sets. The pair (f, F) is *convex-like* on X if

$$(\exists \alpha \in (0,1))(\forall x_1, x_2 \in X)(\exists x_3 \in X),$$
$$f(x_3) \leq \alpha f(x_1) + (1-\alpha) f(x_2)$$

and $(\forall y \in Y)$:

$$F(x_3, y) \leq \alpha F(x_1, y) + (1-\alpha) F(x_2, y).$$

If the pair (f, F) is convex-like on X, there is $x_0 \in X$ with $(\forall y \in Y)\, F(x_0, y) \leq 0$ and if a regularity condition holds then the following statements are equivalent [21]:

$$\forall y \in Y,\ F(x, y) \leq 0 \Longrightarrow f(x) \geq 0,$$
$$(\forall \theta < 0)(\exists \lambda \in \Lambda)(\forall x \in X)$$
$$f(x) + \sum_{y \in Y} \lambda_y F(x, y) > \theta,$$

where Λ is the dual cone of the convex cone of all nonnegative functions on Y. An asymptotic version of the above result holds if the regularity hypothesis is not fulfilled. This extension has been applied to develop Lagrange multiplier type results for minimax problems and constrained optimization problems involving convex-like functions. For related results see [16].

Semidefinite Programming

A useful corollary of the Farkas lemma, which is often used to characterize the *feasibility problem* for linear inequalities, states that exactly one of the following alternatives is true:

i) $\exists x \in \mathbf{R}^n\ a_i^\top x \leq b_i,\ i = 1, \ldots, m$,
ii) $\exists \lambda_i \geq 0\ \sum_{i=1}^m \lambda_i a_i = 0,\ \sum_{i=1}^m b_i \lambda_i = -1$.

This form of the Farkas lemma has also attracted various extensions to nonlinear systems, including *sublinear and DSL systems* [20] with the view to characterize the feasibility of such systems. The feasibility problem, which has been of great interest in *semidefinite programming*, is the problem of determining whether there exists an $x \in \mathbf{R}^n$ such that $Q(x) \succeq 0$, for real symmetric matrices Q_i, $i = 0, \ldots, m$, where $\succeq$ denotes the partial order, i. e. $B \succeq A$ if and only if $A - B$ is positive semidefinite, and $Q(x) = Q_0 - \sum_{i=1}^m x_i Q_i$. However, simple examples show that a direct analog of the alternative does not hold for the semidefinite inequality systems $Q(x) \succeq 0$ without additional hypothesis on Q. A modified dual conditions which characterize solvability of the system $Q(x) \succeq 0$ is given in [28].

See also

▶ Farkas Lemma

References

1. Ben-Israel A (1969) Linear inequalities and inequalities on finite dimensional real or complex vector spaces: A unified theory. J Math Anal Appl 27:367–389
2. Borwein JM (1983) Adjoint process duality. Math Oper Res 8:403–437
3. Borwein JM (1983) A note on the Farkas lemma. Utilitas Math 24:235–241
4. Borwein JM, Wolkowicz H (1982) Characterizations of optimality without constraint qualification for the abstract convex program. Math Program Stud 19:77–100
5. Craven BD (1978) Mathematical programming and control theory. Chapman and Hall, London
6. Craven BD, Koliha JJ (1977) Generalizations of Farkas' theorem. SIAM J Math Anal 8:983–997
7. Farkas J (1901) Theorie der einfachen Ungleichungen. J Reine Angew Math 124:1–27
8. Glover BM (1982) A generalized Farkas lemma with applications to quasidifferentiable programming. Z Oper Res 26:125–141
9. Glover BM, Ishizuka Y, Jeyakumar V, Tuan HD (1996) Complete characterization of global optimality for problems involving the pointwise minimum of sublinear functions. SIAM J Optim 6:362–372
10. Glover BM, Jeyakumar V, Oettli W (1994) Farkas lemma for difference sublinear systems and quasidifferentiable programming. Math Program 63:333–349
11. Glover BM, Jeyakumar V, Rubinov AM (1999) Dual conditions characterizing optimality for convex multi-objective programs. Math Program 84:201–217
12. Goberna MA, Lopez MA, Pastor J (1981) Farkas-Minkowski systems in semi-infinite programming. Appl Math Optim 7:295–308
13. Gwinner J (1987) Corrigendum and addendum to 'Results of Farkas type'. Numer Funct Anal Optim 10:415–418
14. Gwinner J (1987) Results of Farkas type. Numer Funct Anal Optim 9:471–520
15. Ha Ch-W (1979) On systems of convex inequalities. J Math Anal Appl 68:25–34
16. Ills T, Kassay G (1994) Farkas type theorems for generalized convexities. Pure Math Appl 5:225–239
17. Jeyakumar V (1987) A general Farkas lemma and characterization of optimality for a nonsmooth program involving convex processes. J Optim Th Appl 55:449–461
18. Jeyakumar V (1990) Duality and infinite dimensional optimization. Nonlinear Anal Th Methods Appl 15:1111–1122
19. Jeyakumar V, Glover BM (1993) A new version of Farkas' lemma and global convex maximization. Appl Math Lett 6(5):39–43
20. Jeyakumar V, Glover BM (1995) Nonlinear extensions of Farkas' lemma with applications to global optimization and least squares. Math Oper Res 20:818–837
21. Jeyakumar V, Gwinner J (1991) Inequality systems and optimization. J Math Anal Appl 159:51–71
22. Jeyakumar V, Rubinov AM, Glover BM, Ishizuka Y (1996) Inequality systems and global optimization. J Math Anal Appl 202:900–919
23. Kuhn HW, Tucker AW Nonlinear programming, Proc. Second Berkeley Symp. Math. Statist. and Probab., Univ. Calif. Press, Berkeley, CA, pp 481–492
24. Lasserre JB (1997) A Farkas lemma without a standard closure condition. SIAM J Control Optim 35:265–272
25. Mangasarian OL (1969) Nonlinear programming. McGraw-Hill, New York
26. Prékopa A (1980) On the development of optimization theory. Amer Math Monthly 87:527–542
27. Pshenichnyi BN (1971) Necessary conditions for an extremum. M. Dekker, New York
28. Ramana MV (1977) An exact duality theory for semidefinite programming and its complexity implications. Math Program 77:129–162
29. Rubinov AM, Glover BM, Jeyakumar V (1995) A general approach to dual characterizations of solvability of inequality systems with applications. J Convex Anal 2(2):309–344
30. Schirotzek W (1985) On a theorem of Ky Fan and its application to nondifferentiable optimization. Optim 16:353–366
31. Zalinescu C (1978) A generalization of the Farkas lemma applications to convex programming. J Math Anal Appl 66:651–678

Feasible Sequential Quadratic Programming

FSQP

ANDRÉ L. TITS
University Maryland, College Park, USA

MSC2000: 65K05, 65K10, 90C06, 90C30, 90C34

Article Outline

Keywords
Main Ideas
Algorithms
Applications
See also
References

Keywords

Nonlinear programming; Sequential quadratic programming; Successive quadratic programming; Feasible iterates

Feasible sequential quadratic programming (FSQP) refers to a class of *sequential quadratic programming* (SQP) methods that have the additional property that all iterates they construct satisfy the inequality constraints. Thus, for the problem

$$\begin{cases} \min\limits_{x\in\mathbf{R}^n} & f(x) \\ \text{s.t.} & g_j(x) \le 0, \quad j = 1,\dots,m_i, \\ & h_j(x) = 0, \quad j = 1,\dots,m_e, \end{cases} \tag{1}$$

where f, the g_js, and the h_js are smooth, FSQP methods generate a sequence $\{x_k\}$ such that $g_j(x_k) \le 0$ for all j and all k.

From the application's point of view, enforcing *feasibility of the iterates* with respect to inequality constraints is often an important attribute. First, it may be the case that the objective function is simply not defined when certain constraints are violated, for example, with problems involving dynamical systems, in which stability is needed in order for, say, certain steady state errors to be well defined. Second, it may be crucial that a (suboptimal) solution satisfying certain 'hard' constraints be available after a prescribed amount of time has elapsed, too short to allow convergence to the optimal solution. This is the case, for instance in certain real-time control applications. A third situation where feasibility of successive iterates is desirable is in the context of trade-off exploration for design problems. Indeed, trade-offs between 'soft' design specifications cannot be meaningfully explored unless 'hard' specifications are satisfied. From the point of view of numerical algorithms, while maintaining feasibility of successive iterates obviously requires special attention, it also has important beneficial side effects. Namely,

i) the objective function can be forced to decrease at each iteration, and thus can serve as *merit function* in the line search, thereby avoiding the complex issue of choice of an appropriate surrogate merit function; and
ii) as pointed out below, in the context of SQP type methods, the quadratic programs successively constructed all have a nonempty feasible set, which is not the case in general for 'infeasible' methods.

Methods that generate feasible iterates have regained much popularity in recent years with the in-depth investigation of barrier-based *interior point methods*, successively in the context of linear, convex-quadratic, general convex, and nonconvex problems, the class of problems of interest here. Contributions to the latter can be found in the classical book [4] as well as, e. g., in [11] (see [14] for a 'modern' presentation) and [3], and in many recent reports. In those methods, each search direction is typically obtained via the solution of a linear system of equations. FSQP algorithms, on the other hand, being of the SQP type, involve the solution of quadratic programs as subproblems. While they are often impractical for problems with large numbers of variables, SQP-type algorithms are particularly suited to various classes of engineering applications where the number of variables is not too large but evaluations of objective/constraint functions and of their gradients are highly time consuming. Indeed, because these algorithms use quadratic programs as successive models, progress between (expensive) function evaluations is typically significantly better than with algorithms making use of mere linear systems of equations as models.

FSQP algorithms are of the *feasible direction* type in that, while they allow iterates to lie on constraint boundaries, small enough displacements along the search directions they generate always yield feasible points. Indeed, whenever the current iterate lies on or near a nonlinear constraint boundary, the search direction tends to point toward the interior of the feasible set. In that respect FSQP algorithms are analogous to interior point methods. Early feasible direction algorithms (see, e. g., [12,16]) were first order methods, i. e., only first derivatives were used and no attempt was made to accumulate and use second order information. As a consequence, such algorithms converged linearly at best. E. Polak proposed several extensions to these algorithms which take second order information into account when computing the search direction (see [12, Sect. 4.4]). Some of the search directions proposed by Polak can be viewed as modified SQP directions but the fast local convergence rate usually associated with SQP is not preserved. In [1], a feasible SQP algorithm is proposed with emphasis on avoiding costly line searches by making it likely that a full step along the constructed direction is acceptable as a next iterate, even early on in the optimization process. The price paid however is again the loss of fast local convergence.

In this article, we focus on feasible SQP methods which, under appropriate assumptions, preserve the fast rate of convergence of standard SQP methods. Such methods have been considered early on by J.N. Herskovits and L.A.V. Carvalho [5] and in [2,6,8,9,10], and recently also by L. Qi and Z. Wei [13].

Main Ideas

For simplicity, consider the case where only inequality constraints are present, i. e., $m_e = 0$. Suppose that the current estimate x_k for the solution of (1) is feasible, i. e., $g_j(x_k) \le 0$ for all j. The basic SQP direction, d_k^0, is obtained by solving the quadratic programming problem

$$\begin{cases} \min\limits_{d^0} & \frac{1}{2}\left\langle d^0, H_k d^0\right\rangle + \left\langle \nabla f(x_k), d^0\right\rangle \\ \text{s.t.} & g_j(x_k) + \left\langle \nabla g_j(x_k), d^0\right\rangle \le 0, \quad \forall j, \end{cases} \tag{2}$$

where H_k is the Hessian of the Lagrangian, or an estimate thereof. While, in general, QP (2) may be inconsistent, feasibility of x_k, which we seek to enforce, guarantees that it admits a feasible point. Indeed, in particular, $d_k^0 = 0$ is always feasible. Assume that H_k is symmetric positive definite. Then QP (2) has a unique solution d_k^0. It is a simple exercise to show that, in addition, d_k^0 has the interesting property of being a first order descent direction for f at x_k, i. e., $\langle \nabla f(x_k), d_k^0\rangle < 0$.

Suppose now that some constraint, say g_{j_0}, is active at x_k, i. e., $g_{j_0}(x_k) = 0$. Then, if the j_0th constraint is also active in QP (2), then $\langle \nabla g_{j_0}(x_k), d_k^0\rangle = 0$, so that d_k^0 is tangent to the feasible set. Quite possibly, as a result, $g_{j_0}(x_k + td_k^0)$ may be positive for small t, making it difficult, or impossible, to locate a next feasible iterate in direction d_k^0. Thus d_k^0 is not an appropriate search direction for FSQP. However any, however small, amount of tilting of d_k^0 towards the interior of the feasible set makes it a feasible direction. The challenge in FSQP type methods is to tilt d_k^0 enough that a sizable step can be made within the feasible set, but little enough that the fast local convergence properties of sequential quadratic programming are preserved.

With appropriate titling of the basic SQP search direction, and an appropriate line search along the resulting direction d (yielding a next iterate $x_k + t_k d_k$ for some $t_k \in (0, 1]$) a globally convergent feasible SQP algorithm can be constructed. However, the result would be unsatisfactory if the algorithm thus obtained did not exhibit a fast local convergence rate, a property that is generally expected from SQP-type methods. For such rate (in particular, a superlinear rate) to take place, it is critical that a full step of one be eventually taken, i. e., that, when x_k is close to the solution, t_k be equal to one. Here a difficulty already arises in the context of classical (nonfeasible) SQP methods, where it may happen that the line search rule prevents the full step from being taken. This possible conflict between global convergence and fast local convergence is known as the *Maratos effect*. In the context of FSQP methods, this difficulty is compounded by the fact that, in order to be acceptable, in addition to satisfy an appropriate descent criterion, the next iterate must be feasible. This imposes further demands on the Maratos-effect avoidance scheme. Two schemes have been proposed in the literature: second order correction with arc search, and nonmonotone line search.

Algorithms

Following is a simple example of an FSQP algorithm, taken from [10].

```
Parameters: $\alpha \in (0, 1/2)$, $\beta \in (0, 1)$.
Data: $x_0 \in X$, $H_0 = H_0^\top > 0$.
Step 0. Initialization: Set $k = 0$.
Step 1. Computation of a search arc.
    Compute $d_k^0$. If $d_k^0 = 0$, stop.
Compute $d_k^1$ and $\rho_k$ and set
    $d_k = (1 - \rho_k)d_k^0 + \rho_k d_k^1$.
    Compute correction $\tilde{d}_k$.
Step 2. Arc search.
    Compute $t_k$, the first number $t$ in the sequence
    $\{1, \beta, \beta^2, \ldots\}$ satisfying $f(x_k + td_k + t^2\tilde{d}_k) \le$
    $f(x_k) + \alpha t\langle \nabla f(x_k), d_k\rangle$, $g_j(x_k + td_k + t^2\tilde{d}_k) \le 0$,
    $j = 1, \ldots, m_i$.
Step 3. Updates.
    Compute $H_{k+1} = H_{k+1}^\top > 0$.
    Set $x_{k+1} = x_k + t_k d_k + t_k^2\tilde{d}_k$.
    Set $k = k + 1$.
    Go back to Step 1.
```

Algorithm: Simple FSQP

Here, d_k^1 is a feasible direction and a direction of first order descent for f, $\rho \in (0, 1]$ goes to zero fast enough (like $\| d_k^0 \|^2$) when d_k^0 goes to zero, and $\widetilde{d}_k$ is a correction that aims at insuring that the full step of one will be accepted when x_k is close enough to a solution; computation of $\widetilde{d}_k$ involves constraint values at $x_k + d_k$. Under standard assumptions this algorithm is known to generate sequences whose limit points are Karush–Kuhn–Tucker points. Under strengthened assumptions, including the assumption that H_k is updated in such a way that it approximates well, in a certain sense, the Hessian of the Lagrangian as a solution is approached, convergence can be shown to be *Q-superlinear* or *2-step superlinear*. See [10] for details. A refined version of the algorithm of [10] is implemented in the CFSQP/FFSQP software (see [15]). Refinements include the capability to handle equality constraints [6], *minimax* and *constrained minimax problems* and to efficiently handle problems with large numbers of inequality constraints and minimax problems with large numbers of objective functions [8]. Also note that an FSQP method with drastically reduced amount of work per iteration has been recently proposed [7].

Applications

Applications abound where FSQP-type algorithms are of special interest. In particular, as stressed above, such algorithms are particularly appropriate for problems where the number of variables is not too large but functions evaluations are expensive, and feasibility of iterates is desirable (or imperative). Furthermore, problems with a large number of inequality constraints (or minimax problems with large numbers of objective functions), such as finely discretized *semi-infinite optimization* problems, can be handled effectively, making FSQP especially well-suited for problems involving, e. g., time or frequency responses of dynamical systems. Pointers to a large number of applications can be found on the web, at the URL listed above. Application areas include all branches of engineering, medicine, physics, astronomy, economics and finances, to mention but a few.

See also

- ► Optimization with Equilibrium Constraints: A Piecewise SQP Approach
- ► Sequential Quadratic Programming: Interior Point Methods for Distributed Optimal Control Problems
- ► Successive Quadratic Programming
- ► Successive Quadratic Programming: Applications in Distillation Systems
- ► Successive Quadratic Programming: Applications in the Process Industry
- ► Successive Quadratic Programming: Decomposition Methods
- ► Successive Quadratic Programming: Full Space Methods
- ► Successive Quadratic Programming: Solution by Active Sets and Interior Point Methods

References

1. Birge J, Qi L, Wei Z (2000) A variant of the Topkis–Veinott method for solving inequality constrained optimization problems. J Appl Math Optim 41:309–330
2. Bonnans JF, Panier ER, Tits AL, Zhou JL (Aug. 1992) Avoiding the Maratos effect by means of a nonmonotone line search II. Inequality constrained problems – feasible iterates. SIAM J Numer Anal 29(4):1187–1202
3. El-Bakry AS, Tapia RA, Tsuchiya T, Zhang Y (1996) On the formulation and theory of the Newton interior-point method for nonlinear programming. J Optim Th Appl 89:507–541
4. Fiacco AV, McCormick GP (1968) Nonlinear programming: Sequential unconstrained minimization techniques. Wiley New York, New York
5. Herskovits JN, Carvalho LAV (1986) A successive quadratic programming based feasible directions algorithm. In: Bensoussan A, Lions JL (eds) Proc. Seventh Internat. Conf. Analysis and Optimization of Systems – Antibes, June 25-27, 1986, Lecture Notes Control Inform Sci. Springer, Berlin, pp 93–101
6. Lawrence CT, Tits AL (1996) Nonlinear equality constraints in feasible sequential quadratic programming. Optim Methods Softw 6:265–282
7. Lawrence CT, Tits AL (1998) A computationally efficient feasible sequential quadratic programming algorithm. Techn Report Inst Systems Res, Univ Maryland, no. TR 98-46
8. Lawrence CT, Tits AL (1998) Feasible sequential quadratic programming for finely discretized problems from SIP. In: Reemtsen R, Rückmann J-J (eds) Semi-infinite programming. Nonconvex Optim Appl. Kluwer, Dordrecht, pp 159–193
9. Panier ER, Tits AL (1987) A superlinearly convergent feasible method for the solution of inequality constrained optimization problems. SIAM J Control Optim 25(4):934–950
10. Panier ER, Tits AL (1993) On combining feasibility, descent and superlinear convergence in inequality constrained optimization. Math Program 59:261–276
11. Panier ER, Tits AL, Herskovits JN (July 1988) A QP-free, globally convergent, locally superlinearly convergent algorithm for inequality constrained optimization. SIAM J Control Optim 26(4):788–811

12. Polak E (1971) Computational methods in optimization. Acad. Press, New York
13. Qi L, Wei Z (2000) On the constant positive linear independence condition and its application to SQP methods. SIAM J Optim 10:963–981
14. Urban T, Tits AL, Lawrence CT (1998) A primal-dual interior-point method for nonconvex optimization with multiple logarithmic barrier parameters and with strong convergence properties. Techn Report Inst Systems Res Univ Maryland no. TR 98-27
15. WEB: 'www.isr.umd.edu/Labs/CACSE/FSQP/fsqp.html'.
16. Zoutendijk G (1960) Methods of feasible directions. Elsevier, Amsterdam

Feedback Set Problems

FSP

PAOLA FESTA[1], PANOS M. PARDALOS[2], MAURICIO G.C. RESENDE[3]

[1] Dip. Mat. e Inform., University Salerno,
Salerno, Italy

[2] Center for Applied Optim. Department Industrial and Systems Engineering,
University Florida, Gainesville, USA

[3] Information Sci. Res., AT&T Labs Res.,
Florham Park, USA

MSC2000: 90C35

Article Outline

Keywords
Notation and Graph Representation
The Feedback Vertex Set Problem
  Mathematical Model of the Feedback Vertex Set Problem
  Polynomially Solvable Cases
  Approximation Algorithms and Provable Bounds on Undirected Graphs
  Approximation Algorithms and Provable Bounds on Directed Graphs
  Exact Algorithms
The Feedback Arc Set Problem
  Mathematical Model of the Feedback Arc Set Problem
  State of the Art of Feedback Arc Set Problems
A GRASP for Feedback Set Problems
Future Research
Conclusions
See also
References

Keywords

Combinatorial optimization; Feedback set problem; Graph bipartization; Local search; GRASP; FORTRAN subroutines

In recent years (1990) feedback set problems have been the subject of growing interest. They have found applications in many fields, including deadlock prevention [90], program verification [79], and Bayesian inference [2]. Therefore, it is natural that in the past few years there have been intensive efforts on exact and approximation algorithms for these kinds of problems. Exact algorithms have been proposed for solving the problems restricted to special classes of graphs as well as several approximation algorithms with provable bounds for the cases that are not known to be polynomially solvable. The most general feedback set problem consists in finding a minimum-weight (or minimum cardinality) set of vertices (arcs) that meets all cycles in a collection C of cycles in a graph (G, w), where w is a nonnegative function defined on the set of vertices $V(G)$ (on the set of edges $E(G)$). This kind of problem is also known as the *hitting cycle problem*, since one must hit every cycle in C. It generalizes a number of problems, including the *minimum feedback vertex (arc) set problem* in both directed and undirected graphs, the *subset minimum feedback vertex (arc) set problem* and the *graph bipartization problem*, in which one must remove a minimum-weight set of vertices so that the remaining graph is bipartite. In fact, if C is the set of all cycles in G, then the hitting cycle problem is equivalent to the problem of finding the minimum feedback vertex (arc) set in a graph. If we are given a set of special vertices and C is the set of all cycles of an undirected graph G that contains some special vertex, then we have the *subset feedback vertex (arc) set problem* and, finally, if C contains all odd cycles of G, then we have the *graph bipartization problem*. All these problems are also special cases of *vertex (arc) deletion problems*, where one seeks a minimum-weight (or minimum cardinality) set of vertices (arcs) whose deletion gives a graph satisfying a given property. There are different versions of *feedback set problems*, depending on whether the graph is directed or undirected and/or the vertices (arcs) are weighted or unweighted. See [30] for a complete survey, and [91] for a general *NP*-hardness proof for almost all vertex and arc deletion problems restricted to planar graphs. These results apply to the planar bipartization problem, the planar (directed, undirected, or subset) feedback vertex set problems, already proved to be *NP*-hard [33,46]. Furthermore, it is *NP*-complete for planar graphs with no indegree or outdegree exceeding three [46], general graphs with no indegree or outdegree exceeding two [46], and edge-directed graphs [46].

The scope of this article is to give a complete state-of-art survey of exact and approximation algorithms and to analyze a new practical heuristic method called GRASP for solving both feedback vertex and feedback arc set problems.

Notation and Graph Representation

Throughout this paper, we use the following notation and definitions.

A *graph* $G = (V, E)$ consists of a finite set of vertices $V(G)$, and a set of arcs $E(G) \subseteq V(G) \times V(G)$.

An arc (or edge) $e = (v_1, v_2)$ of a directed graph (digraph) $G = (V, E)$ is an incoming arc to v_2 and an outgoing arc from v_1 and it is incident to both v_1 and v_2. If G is undirected, then e is said to be only incident to v_1 and v_2.

For each vertex $i \in V(G)$, let in(i) and out(i) denote the set of incoming and outgoing edges of i, respectively. They are defined only in case of a digraph G. If G is undirected, we will take into account only the degree $\Delta_G(i)$ of i as the number of edges that are incident to i in G.

$\Delta(G)$ denotes the maximum degree among all vertices of a graph G and it is called the *graph degree*.

A vertex $v \in G$ is called an *endpoint* if it has degree one, a *linkpoint* if it has degree two, while a vertex having degree higher than two is called a *branchpoint*.

A *path P* in G connecting vertex u to vertex v is a sequence of arcs $e_1, \dots, e_r$ in $E(G)$, such that $e_i = (v_i, v_{i+1})$, $i = 1, \dots, r$, with $v_1 = u$ and $v_{r+1} = v$. A *cycle C* in G is a path $C = (v_1, \dots, v_r)$, with $v_1 = v_r$.

A *subgraph* $G' = (V', E')$ of $G = (V, E)$ induced by V' is a graph such that $E' = E \cap (V' \times V')$. A graph G is said to be a *singleton*, if $|V(G)| = 1$. Any graph G can be partitioned into isolated connected components $G_1, \dots, G_k$ and the partition is unique. Similarly, every feedback vertex set V' of G can be partitioned into feedback vertex sets $F_1, \dots, F_k$ such that F_i is a feedback vertex set of G_i. Therefore, following the additive property and denoting by $\mu(G, w)$ the weight of a minimum feedback vertex (arc) set for (G, w), we have:

$$\mu(G, w) = \sum_{i=1}^{k} \mu(G_i, w).$$

The Feedback Vertex Set Problem

Formally, the feedback vertex set problem can be described as follows. Let $G = (V, E)$ be a graph and let w: $V(G) \rightarrow \mathbf{R}^+$ be a weight function defined on the vertices of G. A *feedback vertex set* of G is a subset of vertices $V' \subseteq V(G)$ such that each cycle in G contains at least one vertex in V'. In other words, a feedback vertex set V' is a set of vertices of G such that by removing V' from G along with all the edges incident to V', results in a forest. The weight of a feedback vertex set is the sum of the weights of its vertices, and a *minimum feedback vertex set* of a *weighted graph* (G, w) is a feedback vertex set of G of minimum weight. The weight of a minimum feedback vertex set will be denoted by $\mu(G, w)$. The *minimum weighted feedback vertex set problem* (MWFVS) is to find a minimum feedback vertex set of a given weighted graph (G, w). The special case of identical weights is called the *unweighted feedback vertex set problem* (UFVS).

Mathematical Model of the Feedback Vertex Set Problem

As a covering-type problem, the feedback vertex set problem admits an integer zero-one programming formulation. Given a feedback vertex set V' for a graph (G, w), $G = (V, E)$, and a set of weights $w = \{w(v)\}_{v \in V(G)}$, let $x = \{x_v\}_{v \in V(G)}$ be a binary vector such that $x_v = 1$ if $v \in V'$, and $x_v = 0$ otherwise. Let C be the set of cycles in (G, w). The problem of finding the minimum feedback vertex set of G can be formulated as an integer programming problem as follows:

$$\begin{cases} \min & \displaystyle\sum_{v \in V(G)} w(v) x_v \\ \text{s.t.} & \displaystyle\sum_{v \in V(\Gamma)} x_v \geq 1, \quad \forall \Gamma \in C, \\ & 0 \leq x_v \leq 1 \text{ integer}, \quad v \in V(G). \end{cases}$$

If one denotes by C_v the set of cycles passing through vertex $v \in V(G)$, then the dual of the corresponding linear programming relaxation is a *packing problem*:

$$\begin{cases} \max & \sum\limits_{\Gamma \in C} y_\Gamma \\ \text{s.t.} & \sum\limits_{\Gamma \in C_v} y_\Gamma \le w(v), \quad \forall v \in V(G), \\ & y_\Gamma \ge 0, \quad \forall \Gamma \in C. \end{cases}$$

Polynomially Solvable Cases

Given the *NP*-completeness of the feedback vertex set problem, a recent line of research has focused on identifying the largest class of specially structured graphs on which such problems remain polynomially solvable. A pioneering work is due to A. Shamir [79], who proposed a linear time algorithm to find a feedback vertex set for a reducible flow graph. C. Wang, E. Lloyd, and M. Soffa [90] developed an $O(|E(G)|\cdot|V(G)|^2)$ algorithm for finding a feedback vertex set in the class of graphs known as *cyclically reducible graphs*, which is shown to be unrelated to the class of quasireducible graphs. Although the exact algorithm proposed by G.W. Smith and R.B. Walford [83] has exponential running time in general, it returns an optimal solution in polynomial time for certain types of graphs. A variant of the algorithm, called the *Smith–Walford-one algorithm*, selects only candidate sets F of size one and runs in $O(|E(G)|\cdot|V(G)|^2)$ time. The class of graphs for which it finds a feedback vertex set is called *Smith–Walford one-reducible*. In the study of feedback vertex set problems a set of operations called *contraction operations* has had significant impact. They contract the graph $G(V, E)$, while preserving all the important properties relevant to the minimum feedback vertex set. See [56] for a detailed analysis of these reduction procedures which are important for the following two reasons. First, a class of graphs of increasing size is computed, where the feedback vertex set of each graph can be found exactly. Second, most proposed heuristics and approximation algorithms use the reduction schemes in order to reduce the size of the problem. Another line of research on polynomially solvable cases focuses on other special classes, including *chordal* and *interval graphs*, *permutation graphs*, *convex bipartite graphs*, *cocomparability graphs* and on *meshes* and *toroidal meshes*, *butterflies*, and *toroidal butterflies*. The feedback vertex set on chordal and interval graphs can be viewed as a special instance of the generalized clique cover problem, which is solved in polynomial time on chordal graphs [20,93] and interval graphs [65]. For permutation graphs, an algorithm due to A. Brandstädt and D. Kratsch [8] was improved by Brandstädt [7] to run in $O(|V(G)|^6)$ time. More recently (1994), Y.D. Liang [58] presented an $O(|V(G)|\cdot|E(G)|)$ algorithm for permutation graphs that can be easily extended to *trapezoid graphs* while keeping the same time complexity. On interval graphs, C.L. Lu and C.Y. Tang [61] developed a linear-time algorithm to solve the minimum weighted feedback vertex set problem using dynamic programming. S.R. Coorg and C.P. Rangan [19] present an $O(|V(G)|^4)$ time and $O(|V(G)|^4)$ space exact algorithm for cocomparability graphs, which are a superclass of permutation graphs. More recently, Liang and M.S. Chang [13] developed a polynomial time algorithm, that by exploring the structural properties of a cocomparability graph uses dynamic programming to get a minimum feedback vertex set in $O(|V(G)^2|\ |E(G)|)$ time. A recent (1998) line of research [63] on polynomially solvable cases focuses on special undirected graphs having bounded degree and that are widely used as connection networks, namely mesh, butterfly and *k-dimensional cube connected cycle* (CCC_k).

Approximation Algorithms and Provable Bounds on Undirected Graphs

A $2\log_2|V(G)|$-approximation algorithm for the unweighted minimum feedback vertex set problem on undirected graphs is contained in a lemma due to P. Erdös and L. Posa [25]. This result was improved in [66] to obtain a performance ratio of $O(\sqrt{\log|V(G)|})$. R. Bar-Yeruda, D. Geiger, J. Naor, and R.M. Roth [2] gave an approximation algorithm for the unweighted undirected case having ratio less than or equal to 4 and two approximation algorithms for the weighted undirected case having ratios $4\log_2|V(G)|$ and $2\Delta^2(G)$, respectively. To speedup the algorithm, they show how to preprocess the input valid graph by applying the corresponding undirected versions of the Levy–Lowe reduction transformations. For the feedback vertex set problem in general undirected graphs, two slightly different 2-approximation algorithms are described in [3] and [1]. These algorithms improve the approximation algorithms of [2]. They also can find a loop cutset which, under specific conditions, is guaranteed in the worst case to contain less than four times the number of variables contained in a minimum loop cutset. Subsequently, A. Becker and Geiger [4] applied the same reduction procedure from the loop cutset problem to the minimum weighted feedback vertex set problem of [2], but their result is independent of any condition and is guaranteed in the worst case to contain less than twice the number of variables contained in a minimum loop cutset. They [4] propose two greedy approximation algorithms for finding the minimum feedback vertex set V' in a vertex-weighted undirected graph (G, w), one of them having performance ratio bounded by the constant 2 and complexity $O(m+n \log n)$, where $m = |E(G)|$ and $n = |V(G)|$. In [17], F.A. Chudak, M.X. Goemans, D. Hochbaum, and D.P. Williamson showed how the algorithms due to Becker and Geiger [3] and V. Bafna, P. Berman, and T. Fujito [1] can be explained in terms of the primal-dual method for approximation algorithms that are used to obtain approximation algorithms for network design problems. The primal-dual method starts with an integer programming formulation of the problem under consideration. It then simultaneously builds a feasible integral solution and a feasible solution to the dual of the linear programming relaxation. If it can be shown that the value of these two solutions is within a factor of α, then an α-approximation algorithm is found. The *integrality gap* of an integer program is the worst-case ratio between the optimum value of the integer program and the optimum value of its linear relaxation. Therefore, by applying the primal-dual method it is possible to proof that the integrality gap of the integer program under consideration is bounded. In fact, Chudak et al., after giving a new integer programming formulation of the feedback vertex set problem, provided a proof that its integrality gap is at most 2. They also gave the proofs of some key inequalities needed to prove the correctness of their new integer programming formulation.

Theorem 1 *Let V' denote any feedback vertex set of a graph $G = (V, E)$, $E \neq \emptyset$, let τ denote the cardinality of the smallest feedback vertex set for G, and let E(S) denote the subset of edges that have both endpoints in $S \subseteq V(G)$, $b(S) = |E(S)| - |S|+1$. Then*

$$\sum_{v \in V'} [\Delta_G(v) - 1] \geq b(V(G)), \qquad (1)$$

$$\sum_{v \in V'} \Delta_G(v) \geq b(V(G)) + \tau. \qquad (2)$$

If every vertex in G has degree at least two, and V'_M is any minimal feedback vertex set (i. e. $\forall\ v \in V'_M$, $V'_M \setminus \{v\}$ is not a feedback vertex set), then

$$\sum_{v \in V'_M} \Delta_G(v) \leq 2(b(V(G)) + \tau) - 2. \qquad (3)$$

G. Even, Naor, B. Schieber, and L. Zosin [28] showed that the integrality gap of that integer program for the standard cycle formulation of the feedback vertex set problem is Ω(log n). The new integer programming formulation given in [17] is as follows:

$$\begin{cases} \min & \sum_{v \in V(G)} w(v) x_v \\ s.t. & \sum_{v \in S} (\Delta_S(v) - 1) x_v \geq b(S), \\ & \quad S \subseteq V(G) :\ E(S) \neq \emptyset, \\ & x_v \in \{0, 1\}, \quad v \in V(G). \end{cases}$$

The linear programming relaxation is:

$$\begin{cases} \min & \sum_{v \in V(G)} w(v) x_v \\ s.t. & \sum_{v \in S} (\Delta_S(v) - 1) x_v \geq b(S), \\ & \quad S \subseteq V(G) :\ E(S) \neq \emptyset, \\ & x_v \geq 0, \quad v \in V(, G), \end{cases}$$

and its dual is:

$$\begin{cases} \max & \sum_{S} b(S) y_S \\ s.t. & \sum_{S : v \in S} (\Delta_S(v) - 1) y_S \leq w_v, \quad v \in V(G), \\ & y_S \geq 0, \quad S \subseteq V(G) :\ E(S) \neq \emptyset. \end{cases}$$

For the subset feedback vertex problem, the authors of [28] showed that it can be approximated in polynomial time by a factor of min*{2 $\Delta(G)$, 8 log($|V'|$+1), O(log τ^*)}, where τ^* denotes the value of the optimal fractional solution. In [28] the authors also proposed a technique, called bootstrapping, that enhances the O(log $|V'|$) to a factor of O(log τ^*/β), where β denotes the minimum weight of a vertex. The bootstrapping technique iteratively uses a graph partition algorithm. The output of each iteration is by itself a subset* *feedback vertex set and is used as part of the input of the next iteration. After $O(\log|V'|)$ iterations the algorithm gives as output a subset feedback vertex set having weight at most $O(\tau^* \log \tau^*)$. Even, Naor and Zosin [26] improved this result proposing an 8-approximation algorithm. The main tool that they used in developing their approximation algorithm and its analysis is a new version of multicommodity flow, called relaxed multicommodity flow, a hybrid of multicommodity flow and multiterminal flow, in which there are additional constraints, called intercommodity constraints. For each arc, the authors considered the maximum flow among all the commodities, which is shipped along it. They required that for each vertex $v \in V(G)$ the sum of the maximum flows shipped along its incident arcs be bounded by four times the capacity of v. By considering the multicommodity flow, the vertices for which the intercommodity constraints are tight play an important role from the point of view of the connectivity of the graph. They are called intersatured vertices. The main result of [26] is a theorem that bounds the weight of the vertices that must be intersatured, so as to satisfy a given demand vector by the sum of demands.*

Approximation Algorithms and Provable Bounds on Directed Graphs

In general, problems on undirected graphs are relatively easier to handle than problems on directed graphs, since more graph theory can be utilized. Not surprisingly, the approximation results obtained so far for the undirected version are stronger than those for the directed version. In fact, none of the algorithms referred to in the previous subsections apply to the feedback vertex set problem in directed graphs and, in contrast with the undirected version, no analytical results are known for the directed case. A very recent direction of research on approximation algorithms in the directed version focuses on the complete equivalence among all feedback set (and/or feedback subset) problems and among these and the directed minimum capacity multicut problem in circular networks. An exhaustive description of the procedures that reduce any feedback set problem to any other or any of them to the directed minimum capacity multicut problem and vice versa are formalized and used in [27] to obtain an approximation algorithm for the subset feedback arc set problem of a weighted directed graph $G = (V, E)$, where the interesting cycles to be hit are contained in a set of special vertices $X \subseteq V(G)$, where $|X| = k$. The weight of the feedback arc set found by their approximation algorithm is $O(\tau^* \log^2|X|)$, where τ^* is the weight of an optimal fractional feedback set. Nevertheless, their approach can be used to solve any other feedback set problem as well as the directed minimum capacity multicut problem. Even et al. [27] also proposed an algorithm for approximating the minimum weighted subset vertex set problem in the weighted and directed case, leading to a result that holds for any other feedback set problem as well. This approach is an algorithmic adaptation of a theoretical result due to P.D. Seymour [78], who proved that the integrality gap in the case of the unweighted feedback vertex set problem can be at most $O(\log \tau^* \log\log \tau^*)$, where τ^* is defined as above. Even et al. observe that all existence arguments contained in the proof of Seymour's statement can be made constructive and thus, with some additional operations, an algorithm for the unweighted feedback vertex set problem having an approximation factor of $O(\log \tau^* \log\log \tau^*)$ can be obtained. Further modifications of the algorithm lead to a polynomial time approximation scheme applicable to the weighted problem. In $O(|E(G)|\cdot|V(G)|^2)$ time the algorithm finds a feedback vertex set having weight

$$O\left(\min\{\tau^* \log \tau^* \log\log \tau^*,\\ \tau^* \log|V(G)| \log\log|V(G)|\}\right).$$

All the observations contained in [27] improve the $O(\log^2 |V(G)|)$-approximation algorithm for this case [54]. In the case of directed planar graphs, H. Stamm [86] presented an $O(|V(G)|\log |V(G)|)$-approximation algorithm, whose performance guarantee is bounded by the maximum degree of the graph and an $O(|V(G)|^2)$ time approximation algorithm with performance guarantee no more than the number of cyclic faces in the planar embedding of the graph minus 1. M. Cai, X. Deng, and W. Zang [10] obtained a 2.5-approximation algorithm for the minimum feedback vertex set problem on tournaments, improving the previously known algorithm with performance guarantee of 3 by E. Speckenmeyer [85]. Let H be the triangle-vertex incidence matrix of a tournament T and let e be the all-one vector. In [10], necessary and sufficient conditions are established for the linear system $\{x: Hx \geq e, x \geq 0\}$ to be a totally dual integral system (TDI).

Definition 2 A rational linear system $\{x: Ax \geq b, x \geq 0\}$ is called *totally dual integral*, if the optimization problem max $\{y^{\mathsf{T}} b: y^{\mathsf{T}} A \leq c^{\mathsf{T}}, y \geq 0\}$ has an integral optimum solution y for every integral vector c for which the maximum is finite.

It has been shown that any rational polyhedron P has a TDI system $P = \{x: Ax \leq b\}$ representation with A integral, and that, if P is full-dimension, there is a unique minimal TDI system $P = \{x: Ax \leq b\}$ with A and b integral if and only if P is integral. In [11] the authors have extended this approach to the feedback vertex set problems and the cycle packing problem in *bipartite tournaments*, where a bipartite tournament is an orientation of a complete bipartite graph. For the aforementioned problems they have found strongly polynomial time algorithms, which are a consequence of a min-max relaxation on packing and covering directed cycles.

Exact Algorithms

In contrast to the numerous approximation schemes that have been studied, relatively few exact algorithms for the feedback vertex set problem have been proposed. To our knowledge, the first algorithm to find an exact minimal cardinality FVS is due to Smith and Walford [83], who proposed a particular graph partition technique. Although their algorithm solves the problem in an arbitrary directed graph in exponential running time, it returns an optimal solution in polynomial time for certain types of graphs. Later, exact algorithms of enumerative nature often used the graph reduction procedures to speed up the process. One study, [16], essentially used direct enumeration plus reduction and reported satisfactory computational results for a set of partial scan design test problems. T. Orenstein, Z. Kohavi, and I. Pomeranz [67] proposed a somewhat more involved exact enumerative procedure based on graph reduction and efficient graph partitioning methods. Their algorithm has been designed for identifying a minimum feedback vertex set in a digital circuit and it is efficient in random graphs, even though in cliques or graphs that are 'almost' cliques it has an exponential behavior, since the reduction and partition techniques cannot be applied.

Somewhat surprising, exact algorithms for feedback vertex set based on mathematical programming formulation are quite few. Recently (1996), M. Funke and G. Reinelt [32] considered a special variant of feedback problems, namely the problem of finding a maximum weight node induced acyclic subdigraph. They discussed valid and facet defining inequalities for the associated polytope and developed a polyhedral-based exact algorithm presenting computational results obtained by applying a branch and cut algorithm.

The Feedback Arc Set Problem

Given a graph $G = (V, E)$ and a nonnegative weight function $w: E(G) \rightarrow \mathbf{R}^{+}$ defined on the arcs of G, the feedback arc set problem consists of finding a minimum-weight subset of arcs $E' \subseteq E(G)$ that meets every cycle in a given collection C of cycles in (G, w). As in the vertex case, this leads to the *minimum feedback arc set problem* (MWFAS) in both directed and undirected graphs, the *minimum weighted graph bipartization problem* via arc removals, and so on.

Mathematical Model of the Feedback Arc Set Problem

Given an arc weighted graph (G, w), $G = (V, E)$ and the set C of all cycles in G, the minimum weighted feedback arc set problem can be formulated as the following integer programming problem:

$$\begin{cases} \min & \displaystyle\sum_{e \in E(G)} w(e) x_e \\ \text{s.t.} & \displaystyle\sum_{e \in \Gamma} x_e \geq 1, \quad \forall \Gamma \in C, \\ & x_e \in \{0, 1\}, \quad \forall e \in E(G). \end{cases}$$

In its relaxation, the constraints $x_e \in \{0, 1\}$, $\forall\, e \in E(G)$ are replaced by $x_e \geq 0$, $\forall\, e \in E(G)$, obtaining a fractional feedback arc set. As with the feedback vertex set problem, the feedback arc set problem is a *covering problem* and its (linear programming) dual is called a *packing problem*. In the case of the feedback arc set problem this means assigning a dual variable to all interesting cycles to be hit in the given graph, such that for each arc the sum of the variables corresponding to the interesting cycles passing through that arc is at most the weight of the arc itself.

State of the Art of Feedback Arc Set Problems

Feedback arc set problems tend to be easier than their vertex counterparts, especially for planar graphs. In the directed case feedback vertex and feedback arc set problems are each reducible to one another. Even, Naor, Schieber, and Sudan [27] showed how to perform reductions among feedback set problems and feedback subset problems and vice versa, preserving feasible solutions and their costs. In all reductions, there is a one-to-one correspondence between feasible solutions and their corresponding costs. Therefore, an approximate solution to one problem can be translated to an approximate solution of the other problem reducible to this problem. Because most of the reduction procedures can be performed in linear time, these problems can be viewed as different representations of the same problem. Hence, as feedback vertex sets are reduced into feedback arc sets with the same weight and vice versa, all of these problems are equally hard to approximate. In the literature of feedback set problems most of the proposed algorithms are designed to solve the problem in vertex-weighted graphs. One of the pioneering papers on feedback arc set problems is [76], where it is proved that finding a minimum feedback arc set in an arc-weighted reducible flow graph is as difficult as finding a minimum cut in a flow network. The proposed algorithm has complexity $O(mn^2 \log (n^2/m))$, where $m = |E(G)|$ and $n = |V(G)|$. The algorithm was adapted to solve the problem in the vertex-weighted case. Shamir's linear time algorithm [79], used for the unit-weighted case, cannot be applied to solve the arc-weighted problem, because any reduction between arc and vertex set problems does not preserve the reducibility property. Given a directed graph $G = (V, E)$, a *dijoin* $E' \subseteq E(G)$ is a set of arcs such that the graph $G' = (V, B)$, $B = E \cup \{(v, u)\colon (u, v) \in E'\}$ is strongly connected. Given nonnegative weights w_e, $e \in E(G)$, the minimum-weight dijoin problem is to find the dijoin with minimum weight. The feedback arc set problem in planar digraphs is reducible to the problem of finding a minimum-weight dijoin in the dual graph, which is solvable in polynomial time [39]. Stamm [86] proposed a simple 2-approximation algorithm for the minimum weight dijoin problem by superposing two arborescences. It is interesting to observe that, when translated to the dual graph, all these problems lead to problems of hitting certain cutsets of the dual graph, problems which can be approximated within a ratio of 2 by the primal-dual method. Goemans and Williamson [37] proposed a primal-dual algorithm that finds a 9/4-approximate solution to feedback set problems in planar graphs. The first approximation algorithm for the feedback arc set problem was given in [54]. The approximation factor is $O(\log^2 n)$ in the unweighted case, where n is the number of vertices of the input graph. This bound was obtained by using a $O(\log n)$ approximation algorithm for a directed separator that splits the graph into two approximately equally-sized components, S and $\overline{S}$. This separator can be found by approximating special cuts called *quotient cuts*. This result was improved by Seymour [78], who gave a $O(\log n \log \log n)$-approximation algorithm that solves the linear relaxation of the feedback arc set mathematical model and then interprets the optimal fractional solution x^* as a length function defined on the arcs. Systematically, in a recursive fashion, it uses this length function to delete from the graph G all arcs between S and $\overline{S}$. Note that the linear program can be solved in polynomial time by using the ellipsoid or an interior point algorithm. Hence, the quality of the bound in this approach depends on the way the graph is partitioned. Seymour [78] proved the following lemma:

Lemma 3 *For a given strongly connected digraph $G = (V, E)$, suppose there exists a feasible solution x to the feedback arc set problem. If ϕ is the value of the optimal fractional solution x^*, then there exists a partition $(S, \overline{S})$ such that, for some ϵ, $0 < \epsilon < 1$, the following conditions hold: If $\delta^+(S) = \{(u,v)\colon\ (u,v) \in E(G),\ u \in S,\ v \in \overline{S}\}$ and $\delta^-(S) = \{(v,u)\colon\ (v,u) \in E(G),\ u \in S,\ v \in \overline{S}\}$, then the following is true:*

$$\sum_{e \in E(S)} w(e)x(e) \le \epsilon\phi, \tag{4}$$

$$\sum_{e \in E(\overline{S})} w(e)x(e) \le (1-\epsilon)\phi, \tag{5}$$

and either

$$\sum_{e \in \delta^+(S)} w(e) \le 20\epsilon\phi \log\left(\frac{1}{\epsilon}\right) \log\log\phi \tag{6}$$

or

$$\sum_{e \in \delta^-(S)} w(e) \le 20\epsilon\phi \log\left(\frac{1}{\epsilon}\right) \log\log\phi. \tag{7}$$

Furthermore, the partition $(S, \overline{S})$ *can be found in polynomial time.*

This Lemma admits a constructive proof, [27]. The algorithm in this proof finds a feedback arc set having weight $O(\tau^* \log^2 |X|)$, where X is a special set of vertices defining the cycles to be hit and τ^* is the weight of an optimal fractional feedback set. The idea is to reduce the problem to the directed minimum capacity multicut problem in circular networks and of adapting the undirected *sphere growing technique* described in [35] to directed circular networks. Then the graph is decomposed in the following way. A fractional and optimal solution to the directed feedback set problem induces a distance metric on the set of arcs (on the set of vertices) $E(G)$. The approximation algorithm arbitrarily picks a vertex $v \in X$ and solves the shortest path tree problem rooted at v with respect to the metric induced by the fractional solution. The procedure that finds the shortest path tree defines layers with respect to the source v. Each layer is a directed cut that partitions the graph into two parts. The next step of the approximation algorithm is to choose a directed cut and to add the cut to the feedback set constructed so far. The algorithm continues recursively in each part and ends when the graph does not contain any interesting cycles. The key of the algorithm is the choice of the criterion to select the directed cut that partitions the graph. Even et al. decided to relate the weight of the cut to the cost of the fractional solution. More recently (1996), Even, Naor, Schieber, and Zosin [28] showed that, for any weight function defined on the arcs, the subset feedback arc set problem can be approximated in polynomial time by a factor of two. The approximation algorithm consists of successive computations of minimum cuts. Its approximation factor is estimated by considering the capacities of minimum cuts as flow paths. When new minimum cuts are computed, previous flow paths are updated according to the decomposition of the graph induced by an optimal solution.

A GRASP for Feedback Set Problems

Although the approximation algorithms guarantee a solution of a certain quality, for many practical real world cases, heuristic methods can lead to better solutions in a reasonable amount of CPU time. Metaheuristics, such as genetic algorithms, simulated annealing, greedy randomized adaptive search procedures (GRASP), Lagrangian relaxation, and others have been developed with successful computational performance on a wide range of combinatorial optimization problems. Interestingly, however, feedback vertex set problems seem to be an exception. For this family of problems relatively few practical heuristics have been developed. Furthermore, most of the heuristics that seem to be quite successful computationally are greedy type heuristics or generalized greedy type heuristics (e. g. GRASP). Almost all the efficient heuristics developed so far employ the solution-preserved reduction rules studied in [56]. It has been observed in practice that this group of heuristics greatly reduces the cardinality of the graph not only at the beginning of the algorithm, but also dynamically during the execution of node deletion type heuristics. A recent line of research on heuristic approaches is due to P.M. Pardalos, T. Qian, and M.G.C. Resende [70] where three variants of the so-called *greedy randomized adaptive search procedure* (*GRASP*) metaheuristic are proposed for finding approximate solutions of large instances of the feedback vertex set problem in a digraph. GRASP is a multistart method characterized by two phases: a construction phase and a local search phase, also known as a local improvement phase. During the construction phase a feasible solution is iteratively constructed. One element at time is randomly chosen from a *restricted candidate list* (RCL), whose elements are sorted according to some greedy criterion, and is added to the building feedback vertex set and removed from the graph with all its incident arcs. Since the computed solution, in general, may not be locally optimal with respect to the adopted neighborhood definition, the local search phase tries to improve it. These two phases are iterated and the best solution found is kept as an approximation of the optimal solution. To improve the efficiency of the method, Pardalos et al. incorporated in each iteration of their algorithm solution-preserving graph reduction techniques in their directed version and that can be used also to check if a digraph is acyclic, returning an empty reduced graph in case of positive answer. The authors employed the following three greedy functions used to select the node with the maximum $G(i)$ values:

- $G_A(i) = \text{in}(i) + \text{out}(i)$;
- $G_B(i) = \text{in}(i) * \text{out}(i)$;
- $G_C(i) = \max\{\text{in}(i), \text{out}(i)\}$.

Greedy function G_A assigns equal weight to in- and out-degrees. G_B favors the balance between in- and out-degrees. G_C only considers the largest value of the degrees. As demonstrated in [70], G_B produced the best computational results. GRASP was tested on two randomly generated problem sets, finding the optimal solutions to all the problems in the first set, where the optimal values are known (computed in [32]). Furthermore, this GRASP dominates the pure greedy heuristics in all the test instances with comparable running time. In [31], Fortran subroutines are given for finding approximate solutions of both the directed feedback vertex set problem and the directed feedback arc set problem using GRASP. The subroutines for solving approximately the feedback vertex set problem corresponds to the pseudocode algorithm proposed in [70]. The subroutines for solving approximately the feedback arc set problem uses a linear-time procedure proposed in [27] in order to reduce the given feedback arc set problem instance to an equivalent feedback vertex set problem instance, and then the reduced vertex version problem is solved.

Future Research

As has been pointed out in [38], fast construction heuristics combined with local improvement techniques tailored for special applications have been the 'workhorse' of combinatorial optimization in practice. As the design of efficient construction heuristics and local search procedures will be a key to the effective computational procedure for feedback set problems, new approaches are considered that will lead to higher quality solution. New variants of the classical GRASP approach are considered, called *Reactive GRASP* techniques. The first idea along this line has been due to M. Prais and C.C. Ribeiro [74], who used reactive GRASP to a matrix decomposition problem arising in the context of traffic scheduling in satellite-division-multiple-access systems (SS/TDMA). In the reactive GRASP, the restricted candidate list parameter α is not fixed, but selfadjusted according to the quality of the solution previously found during the search. In more detail, the parameter α is randomly chosen from a set of m predeterminated acceptable values $A = \{\alpha_1, \ldots, \alpha_m\}$. Associated with the choice of α_i there is a probability p_i, initially corresponding to a uniform distribution. During the search phase some information is collected in order to change the discrete set of probabilities $\{p_i\}_{i=1,\ldots,m}$. Several possible strategies can be explored for this update operation. One among them has been proposed by Prais and Ribeiro. It is an *absolute qualification rule*, based on the average value of the solutions obtained with each value of $\alpha = \alpha_i$. Once chosen the updating criterion of the probabilities $\{p_i\}_{i=1,\ldots,\mathrm{m}}$, it is possible to use different values of α at different iterations. Therefore, different restricted candidate lists can be built and eventually different solutions can be constructed, which would never be built by using a single, fixed value of α.

T.A. Feo and Resende have discussed in [29] the effects the parameter α can have on the quality of the solution and, at least analyzing the results obtained by Prais and Ribeiro, it seems that α can have an evident impact on the outcome of a GRASP procedure.

Conclusions

Despite the large body of work on approximation algorithms, computational studies of feedback set problems seem to be still in their embryonic stage. No modern metaheuristics, except the GRASP procedure recently (1996) developed in [70] have ever been applied to the feedback vertex set problem. The size of the general problem that can be handled is still quite limited. It seems that this area of computational research has the greatest potential for progress and impact in the coming years. It has to be also underlined that, since detecting cycles is a relatively expensive operation, the local search of feedback vertex set appears to be even more difficult than other notorious combinatorial problems like the traveling salesperson or set covering problems. Therefore, the design of efficient local search procedures and fast construction heuristics will be a key to the effective computational procedure for feedback set problems.

See also

- ► Generalized Assignment Problem
- ► Graph Coloring
- ► Graph Planarization
- ► Greedy Randomized Adaptive Search Procedures
- ► Quadratic Assignment Problem
- ► Quadratic Semi-assignment Problem

References

1. Bafna V, Berman P, Fujito T (1995) Constant ratio approximations of the weighted feedback vertex set problem for undirected graphs. In: Staples J, Eades P, Katoh N, Moffat A (eds) ISAAC95, Algorithms and Computation. Lecture Notes Computedr Sci. Springer, Berlin, pp 142–151
2. Bar-Yehuda R, Geiger D, Naor J, Roth RM (1998) Approximation algorithms for the vertex feedback set problem with applications to constraint satisfaction and Bayesian inference. SIAM J Comput 27(4):942–959
3. Becker A, Geiger D (1994) Approximation algorithm for the loop cutset problem. 10th Conf. Uncertainty in Artificial Intelligence. Morgan Kaufmann, San Mateo, pp 60–68
4. Becker A, Geiger D (1996) Optimization of Pearl's method of conditioning and greedy-like approximation algorithms for the vertex feedback set problem. Artif Intell 83:167–188
5. Bondy JA, Hopkins G, Staton W (1987) Lower bounds for induced forests in cubic graphs. Canad Math Bull 30:193–199
6. Bovet DP, de Agostino S, Petreschi R (1988) Parallelism and the feedback vertex set problem. Inform Process Lett 28:81–85
7. Brandstädt A (1993) On improved time bounds for permutation graph problems. In: 18th Workshop on Graph-theoretic concepts in computer science, vol 657. In: Lecture Notes Computer Sci, vol 657. Wiesbaden-Naurod and Springer, Berlin, pp 1–10
8. Brandstädt A, Kratsch D (1985) On the restriction of some NP-complete graph problems to permutation graphs. In: Budach L (ed) Fundamentals of Computing Theory. Lecture Notes Computer Sci. Springer, Berlin, pp 53–62
9. Breuer MA, Gupta R (1989) BALLAST: A methodology for partial scan design. 19th Internat Symposium on Fault-Tolerant Computing, pp 118–125
10. Cai M, Deng X, Zang W (1998) A TDI system and its application to approximation algorithm. 39th Annual Symposium on Foundations of Computer Sci
11. Cai M, Deng X, Zang W (1999) A min-max theorem on feedback vertex sets. Integer Programming and Combinatorial Optimization. Proc 7th Internat IPCO Conf. In: Lecture Notes Computer Sci. Springer, Berlin
12. Chakradhar S, Balakrishnan A, Agrawal V (1994) An exact algorithm for selecting partial scan flip-flops. unpublished
13. Chang MS, Liang YD (1997) Minimum feedback vertex sets in cocomparability graphs and convex bipartite graphs. Acta Informatica 34:337–346
14. Charon I, Guenoche A, Hudry O, Wairgard F (1997) New results on the computation of median orders. Discret Math 165/166:139–153
15. Chen R, Guo X, Zhang F (1988) The z-transformation graphs of perfect matchings of hexagonal system. Discret Math 72:405–415
16. Cheng KT, Agrawal VD (1990) A partial scan method for sequential circuits with feedback. IEEE Trans Comput 39(4):544–548
17. Chudak FA, Goemans MX, Hochbaum D, Williamson DP (1998) A primal-dual interpretation of two 2-approximation algorithms for the feedback vertex set problem in undirected graphs. Oper Res Lett 22:111–118
18. Chvátal V (1979) A greedy heuristic for the set covering problem. Math Oper Res 4:233–235
19. Coorg SR, Rangan CP (1995) Feedback vertex set on cocomparability graphs. Networks 26:101–111
20. Corneil DG, Fonlupt J (1988) The complexity of generalized clique covering. Discrete Appl Math 22:109–118
21. Dechter R (1990) Enhancement schemes for constraint processing: Backjumping, learning, and cutset decomposition. Artif Intell 41:273–312
22. Dechter R, Pearl J (1987) The cycle cutset method for improving search performance in AI. In: 3rd IEEE on AI Applications
23. Donald J, Elwin J, Hager R, Salamon P (1995) A bad example for the minimum feedback vertex set problem. IEEE Trans Circuits and Systems 32:491–493
24. Downey RG, Fellows MR (1995) Fixed-parameter tractability and completeness I: Basic results. SIAM J Comput 24:873–921
25. Erdös P, Posa L (1962) On the maximal number of disjoint circiuts of a graph. Publ Math Debrecen 9:3–12
26. Even G, Naor JS, Zosin L. An 8-approximation algorithm for the subset feedback vertex problem proposed a 8-approximation algorithm
27. Even G, Naor S, Schieber B, Sudan M (1998) Approximating minimum feedback sets and multicuts in directed graphs. Algorithmica 20:151–174
28. Even G, Naor S, Schieber B, Zosin L (1996) Approximating minimum subset feedback sets in undirected graphs, with applications. 4th Israel Symposium on Theory of Computing and Systems, pp 78–88
29. Feo TA, Resende MG (1995) Greedy randomized adaptive search procedures. J Global Optim 6:109–133
30. Festa P, Pardalos PM, Resende MGC (1999) Feedback set problems. In: Du D-Z, Pardalos PM (eds) Handbook Combinatorial Optim, vol 4, pp 209–258
31. Festa P, Pardalos PM, Resende MGC (1999) Fortran subroutines for approximate solution of feedback vertex set problems using GRASP. AT&T Lab Res, Florham Park
32. Funke M, Reinelt G (1996) A polyhedral approach to the feedback vertex set problem. unpublished
33. Garey MR, Johnson DS (1979) Computers and intractability: A guide to the theory of NP-completeness. Freeman, New York
34. Garey MR, Tarjan RE (1978) A linear-time algorithm for finding all feedback vertices. Inform Process Lett 7:274–276
35. Garg N, Vazirani VV, Yannakakis M (1996) Approximate max-flow min-(multi) cut theorems and their applications. SIAM J Comput 25(2):235–251

36. Gavril F (1977) Some NP-complete problems on graphs. 11th Conf Inform Sci and Systems, Johns Hopkins Univ Press, Baltimore, pp 91–95
37. Goemans MX, Williamson DP (1996) Primal-dual approximation algorithms for feedback problems in planar graphs. 5th MPS Conf Integer Programming and Combinatorial Optimization (IPCO), pp 147–161
38. Grötschel M, Lovász L (1993) Combinatorial optimization: A survey. Techn Report, DIMACS Rutgers Univ 29
39. Grötschel M, Lovász L, Schrijver A (1988) Geometric algorithms and combinatorial optimization. Springer, Berlin, pp 253–254
40. Harary F, Klein DJ, Zivkovic TP (1991) Graphical properties of polyhexes: Perfect matching vector and forcing. J Math Chem 6:295–306
41. Hochbaum D (1982) Approximation algorithms for set covering and vertex cover problem. SIAM J Comput 11(3):555–556
42. Hu TC (1963) Multi-commodity network flows. Oper Res 11:344–360
43. Isaak G (1995) Tournaments as feedback arc sets. Electronic J Combin 20(2):1–19
44. Johnson DS (1974) Approximation algorithms for combinatorial problems. J Comput Syst Sci 9:256–278
45. Johnson DB (1975) Finding all the elementary circuits of a directed graph. SIAM J Comput 4(1):77–84
46. Karp RM (1972) Reducibility among combinatorial problems. In: Miller RE, Thatcher JW (eds) Complexity Of Computer Computations. Plenum, New York, pp 85–103
47. Kevorkian AK (1980) General topological results on the construction of a minimum essential set of a directed graph. IEEE Trans Circuits and Systems 27:293–304
48. Kim H, Perl J. A computational model for combined causal and diagnostic reasoning in inference systems. 8th IJCAI, Morgan Kaufmann, San Mateo, pp 190–193
49. Klein DJ, Randić M (1987) Innate degree of freedom of a graph. J Comput Chem 8:516–521
50. Klein DJ, Zivković TP, Valenti R (1991) Topological long-range order for resonating-valance-bond structures. Phys Rev B 43A:723–727
51. Kunzmann A, Wunderlich HJ (1990) An analytical approach to the partial scan problem. J Electronic Testing: Th Appl 1:163–174
52. Lauritzen SL, Spiegelhalter DJ (1988) Local computations with probabilities on graphical structures and their application to expert systems (with discussion). J Royal Statist Soc B 50:157–224
53. Lee D, Reedy S (1990) On determining scan flip-flops in partial scan designs. Internat Conf Computer Aided Design, pp 322–325
54. Leighton T, Rao S (1988) An approximate max-flow mincut theorem for uniform multicommodity flow problems with applications to approximation algorithms. 29th Annual Symposium on Fundations of Computer Sci, pp 422–431
55. Lempel A, Cederbaum I (1966) Minimum feedback arc and vertex sets of a directed graph. IEEE Trans Circuit Theory CT-13:399–403
56. Levy H, Lowe L (1988) A contraction algorithm for finding small cycle cutsets. J Algorithms 9:470–493
57. Li X, Zhang F (1995) Hexagonal systems with forcing edges. Discret Math 140:253–263
58. Liang YD (1994) On the feedback vertex set problem in permutation graphs. Inform Process Lett 52:123–129
59. Liu J, Zhao C (1996) A new bound on the feedback vertex sets in cubic graphs. Discret Math 148:119–131
60. Lloyd EL, Soffa ML, Wang CC (1988) On locating minimum feedback vertex sets. J Comput Syst Sci 37:292–311
61. LuChin Lung, Tang Chuan Yi (1997) A linear-time algorithm for the weighted feedback vertex problem on interval graphs. Inform Process Lett 61:107–111
62. Lucchesi CL, Younger DH (1978) A minimax theorem for directed graphs. J London Math Soc 17:369–374
63. Luccio FL (1998) Almost exact minimum feedback vertex set in meshes and butterflies. Inform Process Lett 66:59–64
64. Lund C, Yannakakis M (1993) On the hardness of approximating minimization problems. 25th ACM Symp on Theory Of Computing, pp 286–293
65. Marathe MV, Pandu Rangan C, Ravi R (1992) Efficient algorithms for generalized clique covering on interval graphs. Discrete Appl Math 39:87–93
66. Monien B, Schultz R (1981) Four approximation algorithms for the feedback vertex set problems. 7th Conf Graph Theoretic Concepts of Computer Sci. Hauser, pp 315–326
67. Orenstein T, Kohavi Z, Pomeranz I (1995) An optimal algorithm for cycle breaking in directed graphs. J Electronic Testing: Th Appl 7:71–81
68. Pachter L, Kim P (1998) Forcing matchings on square grids. Discret Math 190:287–294
69. Papadimitriou C, Yannakakis M (1988) Optimization, approximation and complexity classes. 20th Annual ACM Symp on Theory of Computing, pp 251–277
70. Pardalos PM, Qian T, Resende MGC (1999) A greedy randomized adaptive search procedure for feedback vertex set. J Combin Optim 2:399–412
71. Peleg D (1996) Local majority voting, small coalitions, and controlling monopolies in graphs: A review. 3rd Colloq Structural Information and Communication Complexity, pp 152–169
72. Peleg D (1997) Size bounds for dynamic monopolies. 4th Colloquium on Structural Information and Communication Complexity, Carleton Univ Press, Ottawa, pp 165–175
73. Perl J (1986) Fusion, propagation and structuring in belief networks. Artif Intell 29:241–288
74. Prais M, Ribeiro CC. Reactive GRASP: An application to a matrix decomposition problem in TDMA traffic assignment
75. Qian T, Ye Y, Pardalos PM (1995) A pseudo-ϵ approximation algorithm for feedback vertex set. In: Floudas CA,

Pardalos PM (eds) Recent Advances in Global Optimization. Kluwer, Dordrecht, pp 341–351

76. Ramachandran V (1988) Finding a minimum feedback arc set in reducible flow graphs. J Algorithms 9:299–313
77. Rosen B (1982) Robust linear algorithms for cutsets. J Algorithms 3:205–217
78. Seymour PD (1995) Packing directed circuits fractionally. Combinatorica 15:281–288
79. Shamir A (1979) A linear time algorithm for finding minimum cutsets in reduced graphs. SIAM J Comput 8(4):645–655
80. Shatcher RD, Andersen SK, Szolovits P (1994) Global conditioning for probabilistic inference in belief networks. In: 10 Conf Uncertainty in AI, pp 514–522
81. Shaw AC (1974) The logical design of operating systems. Prentice-Hall, Upper Saddle River
82. Simovici DA, Grigoras G (1979) Even initial feedback vertex set problem is NP-complete. Inform Process Lett 8:64–66
83. Smith GW, Walford RB (1975) The identification of a minimal feedback vertex set of a directed graph. IEEE Trans Circuits and Systems CAS-22(1):9–14
84. Speckenmeyer E (1988) On feedback vertex sets and nonseparating independent sets in cubic graphs. J Graph Theory 12:405–412
85. Speckenmeyer E (1989) On feedback problems in digraphs. Lecture Notes Computer Sci, vol 411. Springer, Berlin, pp 218–231
86. Stamm H (1990) On feedback problems in a planar digraph. In: Möhring R (ed) Graph-Theoretic Concepts in Computer Sci. Lecture Notes Computer Sci, vol 484. Springer, Berlin, pp 79–89
87. Tarjan RE (1972) Depth first search and linear graph algorithms. SIAM J Comput 1:146–160
88. Ueno S, Kajitani Y, Gotoh S (1988) On the nonseparating independent set problem and feedback set problem for graphs with no vertex degree exceeding three. Discret Math 72:355–360
89. Vazirani V. Approximation algorithms. Manuscript College of Computing, Georgia Inst Techn
90. Wang C, Lloyd E, Soffa M (1985) Feedback vertex sets and cyclically reducible graphs. J ACM 32(2):296–313
91. Yannakakis M (1978) Node and edge-deletion NP-complete problems. 10th Annual ACM Symp Theory of Computing, pp 253–264
92. Yannakakis M (Feb. 1994) Some open problems in approximation. Second Italian Conf Algorithm and Complexity, CIAC'94, pp 33–39
93. Yannakakis M, Gavril F (1987) The maximum k-colorable subgraph problem for chordal graphs. Inform Process Lett 24:133–137
94. Younger DH (1963) Minimum feedback arc set for a directed graph. IEEE Trans Circuit Theory CT-10:238–245
95. Zheng M, Lu X (1990) On the maximum induced forests of a connected cubic graph without triangles. Discret Math 85:89–96

Fejér Monotonicity in Convex Optimization

Patrick L. Combettes[1,2]

[1] Lab. d'Anal. Numér., University Pierre et Marie Curie, Paris, France

[2] City College and Graduate School City, University New York, New York, USA

MSC2000: 47H05, 90C25, 90C55, 65J15, 90C25

Article Outline

Keywords
Notation and Assumptions
Basic Convergence Properties
  Weak Convergence
  Strong Convergence
  Linear Convergence
Geometric Construction
Applications
  Fixed Points of Nonlinear Operators
  Zeros of Monotone Maps
  Zeros of the Sum of Two Monotone Maps
  Variational Inequalities
  Differentiable Optimization
  Convex Feasibility Problems
  Nondifferentiable Optimization
  Inconsistent Convex Feasibility Problems
Historical Notes and Comments
See also
References

Keywords

Fejér monotonicity; Convex programming; Convergence

Let S be a nonempty closed and convex set in a real Hilbert space $\mathcal{H}$ with norm $\|\cdot\|$. A sequence $(x_n)_{n\geq 0}$ of points in $\mathcal{H}$ is said to be *Fejér monotone* with respect to S (or simply *S-Fejérian*) if

$$\forall \overline{x} \in S \ \forall n \in \mathbb{N}: \ \|x_{n+1} - \overline{x}\| \leq \|x_n - \overline{x}\| . \qquad (1)$$

In words, each point in the sequence is not further from any point in S than its predecessor. Given $x_0 \in \mathcal{H}$, a typical example of S-Fejérian sequence is that generated by the algorithm

$$\forall n \in \mathbb{N}: \ x_{n+1} = Tx_n ,$$

where $T{:}\mathcal{H} \to \mathcal{H}$ is a nonexpansive operator, i. e.,

$$\forall (x, y) \in \mathcal{H}^2\colon\ \ \|Tx - Ty\| \le \|x - y\|\,, \tag{2}$$

with nonempty fixed point set S. Under suitable assumptions, the sequence of successive approximations $(x_n)_{n\ge 0}$ converges to a point in S [20].

In convex optimization, one frequently encounters algorithms whose orbits $(x_n)_{n\ge 0}$ are Fejér monotone with respect to the solution set. In order to simplify and standardize the convergence proofs of this broad class of algorithms, it is important to investigate the notion of Fejér monotonicity and to bring out some general convergence principles. These are precisely the objectives of the present article.

Notation and Assumptions

Throughout, the sequence $(x_n)_{n\ge 0}$ is Fejér monotone with respect to a nonempty closed and convex set S in a real Hilbert space $\mathcal{H}$ with scalar product $\langle\cdot|\cdot\rangle$, norm $\|\cdot\|$, and distance d. For every $n \in \mathbf{N}$, p_n denotes be the projection of x_n onto S, i. e., the unique point $p_n \in S$ such that $\|x_n - p_n\| = d(x_n, S)$. Recall that p_n is characterized by the variational inequality

$$\forall \overline{x} \in S\colon\ \ \langle \overline{x} - p_n | x_n - p_n\rangle \le 0. \tag{3}$$

The expressions $x_n \rightharpoonup x$ and $x_n \to x$ denote respectively the weak and strong convergence of $(x_n)_{n\ge 0}$ to x. $\mathfrak{W}$ and $\mathfrak{S}$ denotes respectively the sets of weak and strong cluster points of $(x_n)_{n\ge 0}$. Finally, Id denotes the identity operator on $\mathcal{H}$.

Basic Convergence Properties

By way of preamble, some immediate consequences of (1) are stated below.

Proposition 1 *The following assertions hold.*

i) $(x_n)_{n\ge 0}$ *is bounded.*

ii) $\forall\, \overline{x} \in S\colon\ (\|x_n - \overline{x}\|)_{n\ge 0}$ *converges.*

iii) $(d(x_n, S))_{n\ge 0}$ *is nonincreasing.*

iv) $\forall\, \overline{x} \in S\colon\ x_n \to \overline{x}$ *if and only if* $\varliminf \|x_n - \overline{x}\| = 0$ *if and only if* $S \cap \mathfrak{S} \ne \emptyset$.

Weak Convergence

In general, Fejér monotone sequences do not converge, even weakly (consider for instance the $\{0\}$-Fejérian sequence $((-1)^n x_0)_{n\ge 0}$ with $x_0 \ne 0$). By virtue of Proposition 1i), $\mathfrak{W} \ne \emptyset$ and a necessary condition for $(x_n)_{n\ge 0}$ to converge weakly to a point in S is $\mathfrak{W} \subset S$. A remarkable consequence of Fejér monotonicity is that this condition is also sufficient. To see this, take y_1 and y_2 in $\mathfrak{W}$, say $x_{k_n} \rightharpoonup y_1$ and $x_{l_n} \rightharpoonup y_2$, and $\overline{x} \in S$. By Proposition 1ii),

$$\lim \|x_{k_n} - \overline{x}\|^2 = \lim \|x_{l_n} - \overline{x}\|^2\,.$$

Therefore, by expanding,

$$\lim \|x_{k_n}\|^2 - \lim \|x_{l_n}\|^2 = 2\,\langle \overline{x} | y_1 - y_2\rangle\,.$$

It follows that

$$S \subset \{x \in \mathcal{H}\colon\ \ \langle x | y_1 - y_2\rangle = \alpha\}\,, \tag{4}$$

where $\alpha = (\lim\|x_{k_n}\|^2 - \|x_{l_n}\|^2)/2$. Thus, $(y_1, y_2) \in S^2 \Rightarrow \alpha = \langle y_1 |\, y_1 - y_2\rangle = \langle y_2 \mid y_1 - y_2\rangle \Rightarrow y_1 = y_2$. Consequently, the bounded sequence $(x_n)_{n\ge 0}$ cannot have more than one weak cluster point in S. This fundamental property will be recorded as:

Proposition 2 $(x_n)_{n\ge 0}$ *converges weakly to a point in* S *if and only if* $\mathfrak{W} \subset S$.

Two additional properties are worth mentioning in connection with weak convergence.

- Let $\overline{\text{aff}} S$ be the closed affine hull of S. If $y_1 \ne y_2$, then (4) asserts that S is contained in a closed affine hyperlane. If $\overline{\text{aff}} S = \mathcal{H}$, $\mathfrak{W}$ reduces to a singleton and $(x_n)_{n\ge 0}$ therefore converges weakly.
- Suppose that $x_n \rightharpoonup \overline{x} \in S$ and let $x \in \mathcal{H}$. Then the identities

  $$\begin{aligned}\forall n \in \mathbb{N}\colon\ \ &\|x_n - x\|^2 \\ &= \|x_n - \overline{x}\|^2 + 2\,\langle x_n - \overline{x} | \overline{x} - x\rangle + \|\overline{x} - x\|^2\end{aligned}$$

  together with Proposition 1ii) imply that $(\|x_n - x\|)_{n\ge 0}$ converges.

Strong Convergence

As evidenced by the classical counterexample of [13], $x_n \rightharpoonup \overline{x} \in S \not\Rightarrow x_n \to \overline{x} \in S$. Accordingly, strong convergence conditions for Fejér monotone sequences must be identified.

First, consider the projected sequence $(p_n)_{n\geq 0}$. It follows from (1) and (3) that for every $(m, n) \in \mathbf{N}^2$

$$\begin{aligned}
&\|p_n - p_{n+m}\|^2 \\
&= \|p_n - x_{n+m}\|^2 + 2\langle p_n - x_{n+m} | x_{n+m} - p_{n+m}\rangle \\
&\quad + \|x_{n+m} - p_{n+m}\|^2 \\
&\leq d(x_n, S)^2 - d(x_{n+m}, S)^2 \\
&\quad + 2\langle p_n - p_{n+m} | x_{n+m} - p_{n+m}\rangle \\
&\leq d(x_n, S)^2 - d(x_{n+m}, S)^2 .
\end{aligned}$$

Consequently, since $(d(x_n, S))_{n\geq 0}$ converges by Proposition 1iii), $(p_n)_{n\geq 0}$ is a Cauchy sequence. This establishes:

Proposition 3 *$(p_n)_{n\geq 0}$ converges strongly.*

This result, which is of interest in its own right, also leads to a simple criterion for the strong convergence of $(x_n)_{n\geq 0}$ to a point in S. Indeed, suppose that $\underline{\lim}\, d(x_n, S) = 0$. Then, thanks to Proposition 1iii), $d(x_n, S) \to 0$, i. e., $x_n - p_n \to 0$. On the other hand, by Proposition 3, $p_n \to \overline{x}$ with $\overline{x} \in S$ since S is closed. One thus obtains:

Proposition 4 *$(x_n)_{n\geq 0}$ converges strongly to a point in S if and only if $\underline{\lim}\, d(x_n, S) = 0$.*

Going back to (4), assume now that $(y_1, y_2) \in \mathfrak{S}^2$. Then $\alpha = (\|y_1\|^2 - \|y_2\|^2)/2$ and (4) therefore becomes

$$\begin{aligned}
S &\subset \left\{x \in \mathcal{H}\colon \left\langle x - \frac{y_1 + y_2}{2} \middle| y_1 - y_2\right\rangle = 0\right\} \\
&= \{x \in \mathcal{H}\colon \|x - y_1\| = \|x - y_2\|\} .
\end{aligned} \tag{5}$$

In words, if $(x_n)_{n\geq 0}$ possesses two distinct strong cluster points y_1 and y_2, S is contained in the closed affine hyperplane whose elements are equidistant from y_1 and y_2. If $\overline{\operatorname{aff}}S = \mathcal{H}$, it results from (5) that $(x_n)_{n\geq 0}$ possesses at most one strong cluster point. This happens in particular when the interior of S is nonempty (Slater condition). In this case, however, a sharper result holds, namely $(x_n)_{n\geq 0}$ converges strongly [22].

Linear Convergence

Proposition 1iii) asserts that $(d(x_n, S))_{n\geq 0}$ is nonincreasing. Assume now that it decreases at a linear rate, say

$$\exists \kappa \in]0,1[\ \ \forall n \in \mathbb{N}\colon \ d(x_{n+1}, S) \leq \kappa d(x_n, S). \tag{6}$$

Then, in view of Proposition 4, $x_n \to \overline{x} \in S$. On the other hand, for every $(m, n) \in \mathbf{N}^2$, (1) yields

$$\begin{aligned}
&\|x_n - x_{n+m}\| \\
&\leq \|x_n - p_n\| + \|x_{n+m} - p_n\| \\
&\leq 2d(x_n, S).
\end{aligned}$$

Thus $\|x_n - \overline{x}\| \leq 2d(x_n, S)$ and one reaches the following conclusion.

Proposition 5 *Suppose that (6) holds. Then $(x_n)_{n\geq 0}$ converges linearly to a point $\overline{x} \in S$: $\forall n \in \mathbb{N}\colon \ \|x_n - \overline{x}\| \leq 2\kappa^n d(x_0, S)$*

Geometric Construction

In order to make the above theoretical convergence results more readily applicable in concrete problems, it will henceforth be assumed that $(x_n)_{n\geq 0}$ has been generated by the following algorithm.

```
0   Take $x_0 \in \mathcal{H}$ and set $n = 0$.
1   Generate a closed affine half-space $H_n$ such
    that $S \subset H_n$.
2   Compute the projection $P_n x_n$ of $x_n$ onto $H_n$
    and take $\lambda_n \in [0, 2]$.
3   Set $x_{n+1} = x_n + \lambda_n(P_n x_n - x_n)$.
4   Set $n = n + 1$ and go to step 1.
```

Fejér Monotonicity in Convex Optimization, Algorithm 1
General Fejérian scheme

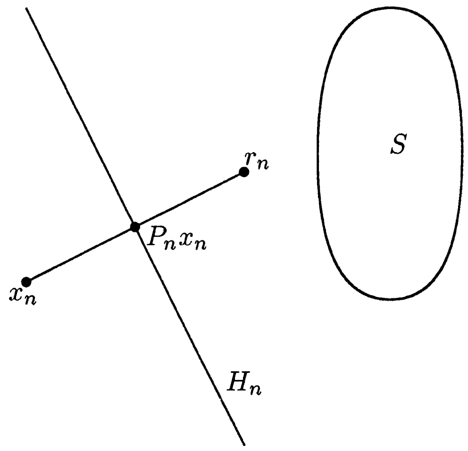
The relaxation parameter λ_n determines the position of the update x_{n+1} on the closed segment between the current iterate x_n and its reflection $r_n = 2P_n x_n - x_n$ with respect to H_n (see Fig. 1.). In some problems, it is possible to significantly accelerate the progression of the iterates towards a solution by proper choice of the relaxation sequence $(\lambda_n)_{n\geq 0}$ [5].

Hereafter, two properties of the relaxation sequence will be considered, namely

$$\sum_{n\geq 0} \lambda_n(2 - \lambda_n) = +\infty \tag{7}$$

and

$$(\lambda_n)_{n\geq 0} \text{ lies in } [\varepsilon, 2 - \varepsilon], \text{ where } \varepsilon \in]0, 1[. \tag{8}$$



Fejér Monotonicity in Convex Optimization, Figure 1
A Fejérian iteration

Now fix $\overline{x} \in S$. Then, for every $n \in \mathbf{N}$,

$$\begin{aligned} &\|x_{n+1} - \overline{x}\|^2 \\ &= \|x_n - \overline{x}\|^2 + \lambda_n^2 \|P_n x_n - x_n\|^2 \\ &\quad + 2\lambda_n \langle x_n - \overline{x} | P_n x_n - x_n \rangle \\ &\leq \|x_n - \overline{x}\|^2 - \lambda_n (2 - \lambda_n) d(x_n, H_n)^2 . \end{aligned} \tag{9}$$

Consequently, $(x_n)_{n \geq 0}$ is S-Fejérian and

$$\sum_{n \geq 0} \lambda_n (2 - \lambda_n) d(x_n, H_n)^2 < +\infty . \tag{10}$$

Furthermore, if $(\lambda_n)_{n \geq 0}$ lies in $[0, 2 - \varepsilon]$ for some $\varepsilon \in]0, 1[$, then the series $\sum_{n \geq 0} \|x_{n+1} - x_n\|^2$ and $\sum_{n \geq 0} \langle \overline{x} - x_n | x_{n+1} - x_n \rangle$ converge [6,15].

In view of (10), the next two convergence results are immediate consequences of Proposition 2 and 4, respectively.

Proposition 6 *$(x_n)_{n \geq 0}$ converges weakly to a point in S if one of the conditions below is fulfilled.*
i) (10) $\Rightarrow \mathfrak{W} \subset S$.
ii) (7) is in force and $\underline{\lim}\, d(x_n, H_n) = 0 \Rightarrow \mathfrak{W} \subset S$.
iii) (8) is in force and $\sum_{n \geq 0} d(x_n, H_n)^2 < + \infty \Rightarrow \mathfrak{W} \subset S$.

Proposition 7 *$(x_n)_{n \geq 0}$ converges strongly to a point in S if one of the conditions below is fulfilled.*
i) (10) $\Rightarrow \underline{\lim}\, d(x_n, S) = 0$.
ii) (7) is in force and $\underline{\lim}\, d(x_n, H_n) = 0 \Rightarrow \underline{\lim}\, d(x_n, S) = 0$.
iii) (8) is in force and $\sum_{n \geq 0} d(x_n, H_n)^2 < + \infty \Rightarrow \underline{\lim}\, d(x_n, S) = 0$.

To investigate linear convergence, assume that

$$\exists \eta \in]0, 1[\ \ \forall n \in \mathbb{N} : \ d(x_n, H_n) \geq \eta d(x_n, S) \tag{11}$$

and that (8) holds. Then $\overline{x} = p_n$ in (11) supplies

$$\begin{aligned} d(x_{n+1}, S)^2 &\leq \|x_{n+1} - p_n\|^2 \\ &\leq d(x_n, S)^2 - \varepsilon^2 d(x_n, H_n)^2 \\ &\leq (1 - \varepsilon^2 \eta^2) d(x_n, S)^2 . \end{aligned}$$

Whence, Proposition 5 yields:

Proposition 8 *Suppose that (8) and (11) hold. Then $(x_n)_{n \geq 0}$ converges linearly to a point $\overline{x} \in S$: $\forall n \in \mathbb{N}: \ \|x_n - \overline{x}\| \leq 2\kappa^n d(x_0, S)$ with $\kappa = (1 - \varepsilon^2 \eta^2)^{½}$.*

Applications

Several convex optimization methods are now presented. They are shown to be Fejér monotone and their convergence is established on the basis of the general results stated above. For brevity, only weak convergence is considered; however, strong and linear convergence results can be derived in a like manner under suitable assumptions. In each problem, the solution set S is assumed to be nonempty.

Fixed Points of Nonlinear Operators

For every $n \in \mathbf{N}$, let $T_n \colon \mathcal{H} \to \mathcal{H}$ be a *firmly nonexpansive* operator, i. e.,

$$\begin{aligned} &\forall (x, y) \in \mathcal{H}^2 : \\ &\qquad \langle T_n x - T_n y | x - y \rangle \geq \|T_n x - T_n y\|^2 , \end{aligned} \tag{12}$$

and let Fix $T_n = \{x \in \mathcal{H} \colon T_n x = x\}$ be its fixed point set. The problem under consideration is to find a common *fixed point* of the family $(T_n)_{n \geq 0}$, i. e.,

$$\begin{cases} \text{Find} & \overline{x} \in \mathcal{H} \\ \text{s.t.} & \forall n \in \mathbb{N} : \ T_n \overline{x} = \overline{x} . \end{cases} \tag{13}$$

Let $S = \bigcap_{n \geq 0} \operatorname{Fix} T_n$ and

$$H_n = \{x \in \mathcal{H} : \ \langle x - T_n x_n | x_n - T_n x_n \rangle \leq 0\} .$$

It then follows from (12) that $S \subset \operatorname{Fix} T_n \subset H_n$. Thus, Algorithm 1 takes the following form.

```
0  Take $x_0 \in \mathcal{H}$ and set $n = 0$.
1  Take $\lambda_n \in [0, 2]$.
2  Set $x_{n+1} = x_n + \lambda_n(T_n x_n - x_n)$.
3  Set $n = n + 1$ and go to step 1.
```

Fejér Monotonicity in Convex Optimization, Algorithm 2
Common fixed point

Noting that $d(x_n, \mathrm{H}_n) = \|(\mathrm{Id} - T_n)\, x_n\|$, several convergence results can be derived by direct application of Propositions 6–8. In particular, in the case of a single nonexpansive operator T (see (2)), the algorithm below is pertinent.

```
0  Take $x_0 \in \mathcal{H}$ and set $n = 0$.
1  Take $\lambda_n \in [0, 1]$.
2  Set $x_{n+1} = x_n + \lambda_n(T x_n - x_n)$.
3  Set $n = n + 1$ and go to step 1.
```

Fejér Monotonicity in Convex Optimization, Algorithm 3
Fixed point

Proposition 9 *If $\sum_{n \geq 0} \lambda_n(1 - \lambda_n) = +\infty$, any sequence generated by Algorithm 3 converges weakly to a fixed point of T.*

Indeed, the assignments $T_n \leftarrow (\mathrm{Id} + T)/2$ and $\lambda_n \leftarrow 2\lambda_n$ in Algorithm 2 yield Algorithm 3 as T_n is firmly nonexpansive [3,5] and Fix T_n = Fix T. Next, observe that $(d(x_n, H_n))_{n \geq 0} = (\|(\mathrm{Id} - T)\, x_n\|/2)_{n \geq 0}$ is nonincreasing by (2). Hence, $\underline{\lim}\; d(x_n, H_n) = 0 \Rightarrow (\mathrm{Id} - T)\, x_n \to 0$ and it results from the demiclosedness of $\mathrm{Id} - T$ [20] that $x_{k_n} \rightharpoonup x \Rightarrow (\mathrm{Id} - T)\, x = 0$. Thus, Proposition 9 follows from Proposition 6ii).

Zeros of Monotone Maps

In connection with set-valued maps $A, B\colon \mathcal{H} \rightrightarrows \mathcal{H}$ a few definitions and facts need to be recalled [2,27]. First, A is characterized by its graph $\mathrm{gr}\, A = \{(x, u) \in \mathcal{H}^2\colon u \in Ax\}$. The inverse A^{-1} of A has graph $\{(u, x) \in \mathcal{H}^2\colon (x, u) \in \mathrm{gr}A\}$ and the linear combination $A + \gamma\, B$ ($\gamma \in \mathbf{R}$) has graph

$$\{(x, u + \gamma v)\colon\; (x, u) \in \mathrm{gr}\, A, (x, v) \in \mathrm{gr}\, B\}\,.$$

A is monotone if

$$\forall (x, u) \in \mathrm{gr}\, A\; \forall (y, v) \in \mathrm{gr}\, A\colon \quad \langle x - y | u - v\rangle \geq 0.$$

If A is monotone and if there exists no monotone map $B \neq A$ such that $\mathrm{gr}\, A \subset \mathrm{gr}B$ then A is *maximal monotone*. In this case

- $\mathrm{gr}\; A$ is weakly-strongly closed: for every sequence $((y_n, v_n))_{n \geq 0}$ in $\mathcal{H}^2$

$$\begin{cases} ((y_n, v_n))_{n\geq 0} \text{ in } \mathrm{gr}\, A \\ y_n \overset{n}{\rightharpoonup} y \\ v_n \overset{n}{\to} v \end{cases} \Rightarrow (y, v) \in \mathrm{gr}\, A. \quad (14)$$

- For every $\gamma \in]0, +\infty[$, the resolvent of A, $J^A_\gamma = (\mathrm{Id} + \gamma\, A)^{-1}$, is a single-valued firmly nonexpansive operator defined on $\mathcal{H}$ [17,23].

Of broad interest is the problem of finding a zero of a maximal monotone map $A\colon \mathcal{H} \rightrightarrows \mathcal{H}$ [23], i. e.,

$$\begin{cases} \text{Find} & \overline{x} \in \mathcal{H} \\ \text{s.t.} & 0 \in A\overline{x}. \end{cases} \quad (15)$$

For every $\gamma \in]\,0, +\infty[$, the solution set $S = A^{-1}\, 0$ can be written as $S = \{x \in \mathcal{H}\colon x \in x + \gamma\; Ax\}$ = Fix J^A_γ. Thus, given $(\gamma_n)_{n \geq 0}$ in $]\,0, +\infty[$, the equilibrium problem (15) can be cast in the form of the common fixed point problem (13) with $(T_n)_{n \geq 0} = (J^A_{\gamma_n})_{n \geq 0}$. Algorithm 2 is then known as the (relaxed) *proximal point algorithm* [17,23].

```
0  Take $x_0 \in \mathcal{H}$ and set $n = 0$.
1  Take $\gamma_n \in ]0, +\infty[$ and $\lambda_n \in [0, 2]$.
2  Set $x_{n+1} = x_n + \lambda_n(J^A_{\gamma_n} x_n - x_n)$.
3  Set $n = n + 1$ and go to step 1.
```

Fejér Monotonicity in Convex Optimization, Algorithm 4
Proximal point

Proposition 10 *Suppose that*

$$\begin{cases} (\gamma_n)_{n\geq 0} & \textit{is in } [\varepsilon, +\infty[\\ (\lambda_n)_{n\geq 0} & \textit{is in } [\varepsilon, 2 - \varepsilon] \end{cases} \quad \textit{where } \varepsilon \in\,]0, 1[\,. \quad (16)$$

Then any sequence generated by Algorithm 4 converges weakly to a zero of A.

This result is a consequence of Proposition 6iii). Indeed, for every $n \in \mathbf{N}$, define $y_n = x_n + (x_{n+1} - x_n)/\lambda_n$, $v_n = (x_n - x_{n+1})/(\gamma_n\, \lambda_n)$ and note that $v_n \in Ay_n$. Now suppose $d(x_n, H_n) \to 0$. Then, thanks to (16), $x_{n+1} - x_n$

$\to 0$ and, in turn, $v_n \to 0$ and $y_n - x_n \to 0$. Hence, $x_{k_n} \rightharpoonup x \Rightarrow y_{k_n} \rightharpoonup x \Rightarrow 0 \in Ax$ by (14).

Weak convergence can also be achieved under variants of (16), e. g., $\sum_{n\geq 0} \gamma_n^2 = +\infty$ and $\forall\, n \in \mathbf{N}\colon \lambda_n = 1$ [2]. Such results can be deduced from Proposition 6 as well.

Zeros of the Sum of Two Monotone Maps

Take two maximal monotone maps $A, B\colon \mathcal{H} \rightrightarrows \mathcal{H}$. An extension of (15) that captures a wide body of optimization and applied mathematics problems is [27]

$$\begin{cases} \text{Find} & \overline{x} \in \mathcal{H} \\ \text{s.t.} & 0 \in A\overline{x} + B\overline{x}. \end{cases} \tag{17}$$

In instances when $A + B$ is maximal monotone, one can approach this problem via Algorithm 4. Naturally, for this approach to be numerically viable, the resolvents of $A + B$ should be computable relatively easily. A more widely applicable alternative is to devise an *operator splitting algorithm*, in which the operators A and B are employed in separate steps [16]. Two Fejérian splitting algorithms are described below.

First, suppose that B is (single-valued and) *cocoercive* in the sense that $B^{-1} - \alpha\ \mathrm{Id}$ is monotone for some $\alpha \in]\,0, +\infty[$, i. e.,

$$\forall (x, y) \in \mathcal{H}^2\colon \quad \langle Bx - By \mid x - y\rangle \geq \alpha \left\|Bx - By\right\|^2 . \tag{18}$$

Given $\gamma \in]\,0, 2\alpha]$, it follows from (18) that $\mathrm{Id} - \gamma B$ is nonexpansive. Moreover, the solution set $S = (A + B)^{-1}\, 0$ can be written as $S = \{x \in \mathcal{H}\colon x - \gamma\ Bx \in x + \gamma Ax\} = \mathrm{Fix}\ T$ where $T = J_\gamma^A \circ(\mathrm{Id} - \gamma\ B)$ is nonexpansive as the composition of two nonexpansive operators. Algorithm 3 can then be implemented by alternating a forward step involving B with a backward (proximal) step involving A.

```
0 | Take $\gamma \in ]0, 2\alpha]$, $x_0 \in \mathcal{H}$, and set $n = 0$.
1 | Set $x_{n+1/2} = x_n - \gamma B x_n$ and take $\lambda_n \in [0, 1]$.
2 | Set $x_{n+1} = x_n + \lambda_n (J_\gamma^A x_{n+1/2} - x_n)$.
3 | Set $n = n + 1$ and go to step 1.
```

Fejér Monotonicity in Convex Optimization, Algorithm 5
Forward-backward method

As a corollary of Proposition 9 we obtain:

Proposition 11 *If $\sum_{n\geq 0} \lambda_n(1 - \lambda_n) = +\infty$, any sequence generated by Algorithm 5 converges weakly to a zero of $A + B$.*

The second algorithm is centered around the operator $T = J_\gamma^A \circ (2J_\gamma^B - \mathrm{Id}) + \mathrm{Id} - J_\gamma^B$, where $\gamma \in]0, +\infty[$. This operator possesses two nice properties: it is firmly nonexpansive and $y \in \mathrm{Fix}\ T \Leftrightarrow J_\gamma^B\ y \in (A + B)^{-1}\ 0$ [16]. Whence, by putting $T_n \leftarrow T$ in Algorithm 2, one obtains the *Douglas–Rachford method* [8,16].

```
0 | Take $\gamma \in ]0, +\infty[$, $x_0 \in \mathcal{H}$, and set $n = 0$.
1 | Set $x_{n+1/2} = J_\gamma^B x_n$ and take $\lambda_n \in [0, 2]$.
2 | Set $x_{n+1} = x_n + \lambda_n \big(J_\gamma^A (2x_{n+1/2} - x_n) - x_{n+1/2}\big)$.
3 | Set $n = n + 1$ and go to step 1.
```

Fejér Monotonicity in Convex Optimization, Algorithm 6
Douglas–Rachford method

As in Algorithm 5, B is activated at step 1 and A at step 2. Convergence is established as in Proposition 9

Proposition 12 *If $\sum_{n\geq 0} \lambda_n(2 - \lambda_n) = +\infty$, any sequence generated by Algorithm 6 converges weakly and the image of the weak limit under J_γ^B is a zero of $A + B$.*

Variational Inequalities

Let $B\colon \mathcal{H} \to \mathcal{H}$ be a single-valued maximal monotone operator, let $\varphi\colon \mathcal{H} \to]-\infty, +\infty]$ be a proper, lower-semicontinuous, convex function, and let $\partial\varphi\colon \mathcal{H} \rightrightarrows \mathcal{H}$ be its subdifferential, i. e.,

$$\partial\varphi(x) = \bigcap_{y\in\mathcal{H}} \{u \in \mathcal{H}\colon\ \langle y - x \mid u\rangle + \varphi(x) \leq \varphi(y)\}\,.$$

Then $\partial\varphi$ is maximal monotone [2] and, upon taking $A = \partial\varphi$ in (17), one arrives at the *variational inequality* problem

$$\begin{cases} \text{Find} & \overline{x} \in \mathcal{H} \\ \text{s.t.} & \forall x \in \mathcal{H}\colon \\ & \quad \langle \overline{x} - x \mid B\overline{x}\rangle + \varphi(\overline{x}) \leq \varphi(x). \end{cases} \tag{19}$$

In this context, the resolvent J_γ^A reduces to Moreau's prox mapping [18]

$$\mathrm{prox}_\gamma^\varphi\colon x \mapsto \arg\min_{y\in\mathcal{H}} \varphi(y) + \frac{1}{2\gamma}\left\|y - x\right\|^2 .$$

As a special instance of (17), the variational inequality problem (19) can be solved via the forward-backward method (Algorithm 5) and Proposition 11 then yields:

Proposition 13 *Suppose that (18) is in force. Take* $\gamma \in]0, 2\alpha]$, $x_0 \in \mathcal{H}$, *and let*

$$\forall n \in \mathbb{N}: \quad x_{n+1} = x_n + \lambda_n \left(\operatorname{prox}_\gamma^\varphi (x_n - \gamma B x_n) - x_n \right), \qquad (20)$$

where $(\lambda_n)_{n \geq 0}$ *is in [0, 1] and* $\sum_{n \geq 0} \lambda_n(1 - \lambda_n) = +\infty$. *Then* $(x_n)_{n \geq 0}$ *converges weakly to a solution of (19).*

A noteworthy situation is when $\varphi = \iota_Q$, where ι_Q is the indicator function of a nonempty closed convex set Q, i. e.,

$$\iota_Q : x \mapsto \begin{cases} 0 & \text{if } x \in Q, \\ +\infty & \text{if } x \notin Q. \end{cases} \qquad (21)$$

It follows that $\partial\iota_Q = N_Q$, where N_Q is the normal cone to Q, i. e.,

$$N_Q x = \bigcap_{y \in Q} \{ u \in \mathcal{H} : \ \langle y - x | u \rangle \leq 0 \},$$

if $x \in Q$, and $N_Q x = \emptyset$ otherwise. In addition, (19) reads

$$\begin{cases} \text{Find} & \overline{x} \in Q \\ \text{s.t.} & \forall x \in Q : \ \langle \overline{x} - x | B\overline{x} \rangle \leq 0, \end{cases} \qquad (22)$$

and $\operatorname{prox}_\gamma^{\iota_Q} = P_Q$ is the projector onto Q.

Differentiable Optimization

A standard convex programming problem is to minimize a proper, lower-semicontinuous, convex function $f: \mathcal{H} \to]-\infty, +\infty]$ over a nonempty closed convex set $Q \subset \mathcal{H}$, i. e.,

$$\text{Find} \quad \overline{x} = \arg\min_{x \in Q} f(x). \qquad (23)$$

In view of (21), (23) is equivalent to finding a global minimizer of $\iota_Q + f$, i. e., by Fermat's rule, to finding a zero of $\partial(\iota_Q + f)$. If 0 lies in the interior of $Q - \{x \in \mathcal{H}: f(x) < +\infty\}$, then $\partial(\iota_Q + f) = \partial\iota_Q + \partial f$ [2] and (23) is therefore of the form (17) with $A = N_Q$ and $B = \partial f$. This occurs in particular when f is finite and continuous at a point in Q.

Now suppose that f is differentiable. Then $\partial f = \{\nabla f\}$ is single-valued and (23) can further be reduced to (22) with $B = \nabla f$. The forward-backward scheme (20) then becomes the *projected gradient algorithm*

$$\forall n \in \mathbb{N}: \quad x_{n+1} = x_n + \lambda_n \left(P_Q \left(x_n - \gamma \nabla f(x_n) \right) - x_n \right).$$

Proposition 13 provides conditions for weak convergence to a minimizer of f over Q.

Convex Feasibility Problems

Given a family $(S_i)_{i \in I}$ of intersecting nonempty closed and convex subsets of $\mathcal{H}$, the *convex feasibility problem* reads [3,5,6,15]

$$\text{Find} \quad \overline{x} \in S = \bigcap_{i \in I} S_i. \qquad (24)$$

At iteration n, select a nonempty finite index set $I_n \subset I$ and, for every $i \in I_n$, let $p_{i,n}$ be an approximate projection of x_n onto S_i, i. e., the projection of x_n onto a closed affine half-space $H_{i,n}$ containing S_i. Then

$$H_{i,n} = \{ x \in \mathcal{H} : \ \langle x - p_{i,n} | x_n - p_{i,n} \rangle \leq 0 \}.$$

Let

$$H_n = \left\{ x \in \mathcal{H} : \sum_{i \in I_n} w_{i,n} \langle x - p_{i,n} | x_n - p_{i,n} \rangle \leq 0 \right\}$$

where the weights $(w_{i,n})_{i \in I_n}$ are in $]0, 1]$ and satisfy $\sum_{i \in I_n} w_{i,n} = 1$. Then $S \subset \cap_{i \in I_n} S_i \subset \cap_{i \in I_n} H_{i,n} \subset H_n$ and $P_n x_n = x_n + L_n(x_{n+1/2} - x_n)$, where $x_{n+1/2} = \sum_{i \in I_n} w_{i,n} p_{i,n}$ and

$$L_n = \begin{cases} \dfrac{\sum_{i \in I_n} w_{i,n} \| p_{i,n} - x_n \|^2}{\left\| x_{n+\frac{1}{2}} - x_n \right\|^2} & \text{if } \ x_{n+\frac{1}{2}} \neq x_n \\ 1 & \text{else.} \end{cases} \qquad (25)$$

Algorithm 1 then turns into Algorithm 7.

```
0  Take $x_0 \in \mathcal{H}$ and set $n = 0$.
1  Take a nonempty finite set $I_n \subset I$.
2  Compute approximate projections $(p_{i,n})_{i \in I_n}$ of
   $x_n$ onto $(S_i)_{i \in I_n}$.
3  Take $(w_{i,n})_{i \in I_n}$ in $]0,1]$ such that
   $\sum_{i \in I_n} w_{i,n} = 1$.
4  Set $x_{n+1/2} = \sum_{i \in I_n} w_{i,n} p_{i,n}$, $L_n$ as in (25).
5  Take $\lambda_n \in [0, 2L_n]$.
6  Set $x_{n+1} = x_n + \lambda_n(x_{n+1/2} - x_n)$.
7  Set $n = n + 1$ and go to step 1.
```

Fejér Monotonicity in Convex Optimization, Algorithm 7
Convex feasibility

Weak convergence to a point in S follows from Proposition 6 under various assumptions on the control sequence $(I_n)_{n \geq 0}$ and the approximate projections $((p_{i,n})_{i \in I_n})_{n \geq 0}$ [5,6,15].

Nondifferentiable Optimization

Suppose that f is subdifferentiable in (23), i. e.,

$$\forall x \in \mathcal{H}: \ \partial f(x) \neq \emptyset,$$

and that its minimum value $\overline{f}$ over Q is known. Then (23) can be viewed as a special case of (24) with two sets, namely $S_1 = Q$ and $S_2 = \{x \in \mathcal{H}: \ f(x) \leq \overline{f}\}$. Now take

$$H_{2,n} = \left\{x \in \mathcal{H}: \ \langle x - x_n | u_n \rangle \leq \overline{f} - f(x_n)\right\}$$

where $u_n \in \partial f(x_n)$. Then $S_2 \subset H_{2,n}$ and

$$p_{2,n} = \begin{cases} x_n + \dfrac{\overline{f} - f(x_n)}{\|u_n\|^2} u_n & \text{if } x_n \notin S_2 \\ x_n & \text{otherwise} \end{cases}$$

is called a *subgradient projection* of x_n onto S_2 [3,5]. If Algorithm 7 is implemented by alternating a relaxed subgradient projection onto S_2 with an exact projection onto S_1, i. e.,

$$\forall n \in \mathbb{N}: \ x_{n+1} = P_Q\left(x_n + \lambda_n\left(p_{2,n} - x_n\right)\right),$$

one obtains the subgradient projection method of [21]. Weak convergence to a solution of (23) under the assumptions of uniform boundedness of ∂f on bounded sets, $(\lambda_n)_{n \geq 0}$ is in $[0, 2]$, and (8), follows from Proposition 6iii [3,5].

Inconsistent Convex Feasibility Problems

When $\cap_{i \in I} S_i = \emptyset$ and I is finite, (24) can be replaced by the minimization problem

$$\text{Find} \quad \overline{x} = \arg\min_{x \in \mathcal{H}} \frac{1}{2} \sum_{i \in I} w_i d(x, S_i)^2 \qquad (26)$$

where $(w_i)_{i \in I}$ is in $]0,1]$ and $\sum_{i \in I} w_i = 1$. Let $(P_i)_{i \in I}$ be the projectors onto $(S_i)_{i \in I}$, let $T = \sum_{i \in I} w_i P_i$, and let S be the solution set of (26). Then T is firmly nonexpansive and $S = \text{Fix } T$ [5]. By reiterating a previous argument, one obtains:

Proposition 14 *Take $x_0 \in \mathcal{H}$, $(\lambda_n)_{n \geq 0}$ in $[0, 2]$ such that $\sum_{n \geq 0} \lambda_n(2 - \lambda_n) = +\infty$, and let*

$$\forall n \in \mathbb{N}: \ x_{n+1} = x_n + \lambda_n \left(\sum_{i \in I} w_i P_i x_n - x_n \right).$$

Then $(x_n)_{n \geq 0}$ converges weakly to a solution of (26).

Historical Notes and Comments

In 1922, L. Fejér considered the following problem [12]: given a closed subset $S \subset \mathbf{R}^p$ and a point $y \notin S$ can one find a point $x \in \mathbf{R}^p$ such that

$$\forall \overline{x} \in S: \ \|x - \overline{x}\| < \|y - \overline{x}\|.$$

Inspired by this work, T.S. Motzkin and I.J. Schoenberg adopted in their 1954 paper [19] the term Fejér monotone to describe sequences satisfying (1). In this paper (see also [1]), an algorithm was developed to solve systems of linear inequalities in $\mathbf{R}^p$ by successive projections onto the half-spaces defining the polyhedral solution set S. The concept of Fejér monotonicity was shown to be an adequate tool to study convergence of this algorithm. Basic facts such as (5) and (9) can already be found in [19] and [1], respectively.

In the 1960s, I.I. Eremin extended the use of Fejér monotonicity to more general convex problems in Hilbert spaces. A summary of his publications covering the period 1961–1967 is given in [9]. By the end of the 1960s, most results on Fejér monotonicity in Hilbert spaces were essentially known and one can find them scattered in the Soviet literature in the context of specific convex programming problems. Thus, (4) appears in [10], Proposition 2 in [4], Propositions 4 and 5 in [14], and Proposition 8 in [14] and [21]. It should be noted that Proposition 2 has been implicitly rediscovered many times and that it seems to originate in [24].

Recently, Fejér monotonicity has been reserved a featured role in several convex optimization papers [3,6,15,25,26]. It has also proven a valuable tool in more applied disciplines such as biology, economics, and engineering [5,11]. Some extensions of the notion of Féjer monotonicity are discussed in [7].

See also

- ► Generalized Monotone Multivalued Maps
- ► Generalized Monotone Single Valued Maps
- ► Generalized Monotonicity: Applications to Variational Inequalities and Equilibrium Problems

References

1. Agmon S (1954) The relaxation method for linear inequalities. Canad J Math 6:382–392
2. Aubin J-P, Cellina A (1984) Differential inclusions. Springer, Berlin
3. Bauschke HH, Borwein JM (1996) On projection algorithms for solving convex feasibility problems. SIAM Rev 38:367–426
4. Brègman LM (1965) The method of successive projection for finding a common point of convex sets. Soviet Math Dokl 6:688–692
5. Combettes PL (1996) The convex feasibility problem in image recovery. In: Advances in Imaging and Electron Physics, vol 95. Acad, pp 155–270
6. Combettes PL (1997) Hilbertian convex feasibility problem: Convergence of projection methods. Appl Math Optim 35:311–330
7. Combettes PL (2001) Quasi-Fejérian analysis of some optimization algorithms. In: Butnariu D, Censor Y, Reich S (eds) Inherently Parallel Algorithms for Feasibility and Optimization and Their Applications. Elsevier, Amsterdam
8. Eckstein J, Bertsekas DP (1992) On the Douglas–Rachford splitting method and the proximal point algorithm for maximal monotone operators. Math Program 55:293–318
9. Eremin II (1968) Methods of Fejér approximations in convex programming. Math Notes 3:139–149
10. Eremin II (1968) On the speed of convergence in the method of Fejér approximations. Math Notes 4:522–527
11. Eremin II, Mazurov VD (1979) Nonstationary processes of mathematical programming. Nauka, Moscow
12. Fejér L (1922) Uber die Lage der Nullstellen von Polynomen, die aus Minimumforderungen gewisser Art entspringen. Math Ann 85:41–48
13. Genel A, Lindenstrauss J (1975) An example concerning fixed points. Israel J Math 22:81–86
14. Gubin LG, Polyak BT, Raik EV (1967) The method of projections for finding the common point of convex sets. USSR Comput Math Math Phys 7:1–24
15. Kiwiel KC, Łopuch B (1997) Surrogate projection methods for finding fixed points of firmly nonexpansive mappings. SIAM J Optim 7:1084–1102
16. Lions PL, Mercier B (1979) Splitting algorithms for the sum of two nonlinear operators. SIAM J Numer Anal 16:964–979
17. Martinet B (1972) Détermination approchée d'un point fixe d'une application pseudo-contractante. Cas de l'application prox. CR Acad Sci Paris Sér A Math 274:163–165
18. Moreau J-J (1962) Fonctions convexes duales et points proximaux dans un espace Hilbertien. CR Acad Sci Paris Sér A Math 255:2897–2899
19. Motzkin TS, Schoenberg IJ (1954) The relaxation method for linear inequalities. Canad J Math 6:393–404
20. Opial Z (1967) Weak convergence of the sequence of successive approximations for nonexpansive mappings. Bull Amer Math Soc 73:591–597
21. Polyak BT (1969) Minimization of unsmooth functionals. USSR Comput Math Math Phys 9:14–29
22. Raik E (1969) A class of iterative methods with Fejér-monotone sequences. Eesti NSV Tead Akad Toimetised Füü-Mat 18:22–26
23. Rockafellar RT (1976) Monotone operators and the proximal point algorithm. SIAM J Control Optim 14:877–898
24. Schaefer H (1957) Uber die Methode sukzessiver Approximationen. Jahresber Deutsch Math-Verein 59:131–140
25. Schott D (1991) A general iterative scheme with applications to convex optimization and related fields. Optim 22:885–902
26. Schott D (1995) Iterative solution of convex problems by Fejér-monotone methods. Numer Funct Anal Optim 16:1323–1357
27. Zeidler E Nonlinear functional analysis and its applications II/B – Nonlinear monotone operators. Springer, Berlin

Financial Applications of Multicriteria Analysis

CONSTANTIN ZOPOUNIDIS
Department Production Engineering
and Management Financial Engineering Lab. Techn.,
University Crete University Campus, Chania, Greece

MSC2000: 91B06, 91B60

Article Outline

Keywords
Basic Principles of Multicriteria Analysis
Methods
Decision Aid Activity
Multicriteria Character of Financial Problems and Some Real-World Applications
Investment Decision
Portfolio Management
Some Real-World Applications
Venture Capital Investment
Study Context
Multicriteria Method and Results
The Business Failure Risk
Study Context
Multicriteria Method and Results
Concluding Remarks
See also
References

Keywords

Finance; Investment analysis; Multicriteria analysis; Decision support; Applications

The financial decisions of an organization (i. e. firm, bank, insurance company, etc.) are usually considered in the context of optimization. Concerning the case of a firm and for a long term period, one meets two types of decisions: decisions related to the optimal allocation of funds, and decisions related to the optimal financial structure. In the short term, one considers decisions related to the management of working capital, and refers to the optimization of stocks, cash, accounts receivable and short term debts. The financial theory analyzes these decisions (short and long terms), but always from an optimization perspective (for example, theory of cost of capital, portfolio theory, options theory, etc.). This perspective has led some researchers to propose techniques of operations research to solve financial decision problems. The classical modeling of decision problems in operations research consists in formulating an optimization (maximization or minimization) problem under specific constraints. In fact, it is a best choice problem.

However, recently, these financial problems have been examined from a more comprehensive and more realistic perspective which overcomes the restrictive framework of optimization [80,84]. For example, in capital budgeting decision making, K. Bhaskar and P. McNamee [6] pose the following questions:

a) In assessing investment proposals, do the decision makers have a single objective or multiple objectives?
b) If decision makers do have multiple objectives, which are those and what is the priority structure of the objectives?

In another similar study, Bhaskar [5] refers that microeconomic theory has largely adopted a single objective function which is the principle of utility maximization for the consumer unit and profit maximization for firms. To attack the single objective function principle for firms, Bhaskar [5] addresses three categories of criticism:

a) there exist alternatives to the profit maximization approach which are based on equally simple hypotheses and which can better explain reality;
b) the profit maximization or any other equally simple hypothesis is too naive to explain the complex process of decision making;
c) the real-world firms do not have suitable information to enable them to maximize their profits. Furthermore, several other theories of the firm have been postulated which have proposed different objectives than that of the traditional microeconomic theory.

One can cite the revenue maximizing model [3], the manager's utility model [74], the satisficing model [64] and the behavioral models [13].

On the basis of the above remarks it is possible to distinguish three main reasons which have motivated a change of view in the modeling of the financial problems:

1) Formulating the problem in terms of seeking the optimum, financial decision makers (i. e. financial analysts, portfolio managers, investors, etc.) get involved in a very narrow problematic, often irrelevant to the real decision problem.
2) The different decisions (financial ones) are taken by the people (i. e. financial managers) and not by the models; the decision makers get more and more deeply involved in the decision making process and, in order to solve problems, it becomes necessary to take into consideration their preferences, their experiences and their knowledge.

3) For financial decision problems such as the choice of investment projects, the portfolio selection, the evaluation of business failure risk, etc., it seems illusory to speak of optimality, since multiple criteria must be taken into consideration.

In this article, our basic aim is to examine the contribution of the multicriteria analysis to the study and to the solution of some financial decision problems. Section 2 presents the basic principles of multicriteria analysis. The multicriteria character of some financial problems and some real world applications of the multicriteria analysis in the field of financial management are given in section 3. Finally, some discussion and the advantages which resulted by the application of multicriteria analysis in the field of financial management, are given in section 4.

Basic Principles of Multicriteria Analysis

Multicriteria analysis, often called *multiple criteria decision making* (MCDM) by the American School and *multicriteria decision aid* (MCDA) by the European School, is a set of methods which allow the aggregation of several evaluation criteria in order to choose one or more alternatives (i. e. investment projects, financial assets at variable revenue, financial assets at fixed revenue, dynamic firms, etc.). It also deals with the study of the activity of decision aid to a well-identified decision maker (i. e. individual, firm, organization, etc.).

The development of multicriteria decision aid (hence we use this term in the text) began in 1971. Its principal objective is to provide the decision maker with tools in order to enable him to advance in solving a decision problem (for example, the selection of investment projects for a firm), where several, often conflicting multiple criteria must be taken into consideration.

Methods

The specialists in the field distinguish several categories of methods in MCDA. The boundaries between these categories are, of course, rather fuzzy. B. Roy [58] proposes the following three categories of methods:

1) unique synthesis criterion approach disregarding any incomparability;
2) outranking synthesis approach, accepting incomparability; and
3) interactive local judgement approach with trial-error iterations.

In this paper, the classification proposed in [53] is adopted. It distinguishes four categories:

1) multi-objective mathematical programming;
2) multi-attribute utility theory;
3) outranking relations approach; and
4) preference disaggregation approach.

Multi-objective mathematical programming is characterized by the fact that an action (or alternative) a is represented by a vector of real variables $(x_1, \dots, x_l)$. The set A of the feasible solutions is defined by a set of linear constraints: $A = \{x \in \mathbf{R}^l \colon A \cdot X \leq b, x \geq 0\}$ with A a matrix of dimensions $m \times l$ and b a vector-matrix $m \times 1$. The chosen vector must give satisfaction to relatively several numerical criteria, m in number, and noted as $C^1, \dots, C^m$, which are linear functions of x. It is possible to distinguish three different methods inside this approach:

1) the efficient solutions procedure;
2) the goal programming;
3) the compromise programming.

A synthesis of the studies realized on this category of methods can be found in [69,72] and [77].

Multi-attribute utility theory (MAUT) is an extension of the classical utility theory. It seeks to give a representation of the preferences of a decision maker by means of a utility function, aggregating several evaluation criteria: $u(\mathbf{g}) = u(g_1, \dots, g_n)$. In other words, the problem is to choose the action a which maximizes the utility function of the decision maker: $u[\mathbf{g}\,(a)] = \max u[\mathbf{g}\,(a)]$.

The criteria (attributes) can be certain or probabilistic (each $g_i(a)$ is associated with a probability distribution). In general, one can decompose a multicriteria utility function in real functions $u_1, \dots, u_n$ concerning the independence of criteria. Thus, different utility function models are obtained. The most studied form of utility function, from a theoretical point of view, is the additive form:

$$u(g_1, \dots, g_n) = u_1(g_1) + \cdots + u_n(g_n),$$

where $u_1, \dots, u_n$ are the marginal utilities defined on the scales of criteria. For the study of the condition of independence in utility between criteria (substitution rate), one can refer to [34]. The latter and [77] present syntheses of works on the construction of multicriteria utility functions both, under certainty and under uncertainty.

The *outranking relations* approach was developed in Europe with the elaboration of the ELECTRE methods (ELimination Et Choix Traduisant la REalité). The concept of outranking in ELECTRE methods is due to Roy, who is the founder of these methods. The outranking relation allows to conclude that an action $a \in A$ (discrete set) outranks an action $b \in A$ if there are enough arguments to confirm that a is at least as good as b, while there is no essential reason to refute this statement. In the ELECTRE methods the aggregation of criteria requires to define the threshold notions of preference and indifference, concordance and discordance. In fact, a outranks b if there exists a sufficient majority of criteria for which a is better classified than b (concordance) and if the unfavorable deviations for the rest of the criteria (discordance) are not too high. Thus, this modeling can bring into evidence the cases of incomparability when the multicriteria evaluation of two actions is very differentiated. A detailed presentation of all outranking methods can be found in [61,63] and [72].

The approach of the *disaggregation of preferences* is often used in MCDA as a mean for the modeling of the preferences of a decision maker or a group of decision makers. This approach uses the regression methods. The introduction of regression methods in MCDA is effected because of the development of the social judgement theory. Multiple regression can, in general, detect, identify or 'capture' the judgement policy of a decision maker (i. e. disaggregation of the preferences). This one, particularly if it is in relation with a certain number of past decisions, might be the expression of a global preference. The approach by multiple regression is quite close to the MAUT; their differences are placed at the level of obtaining the marginal utilities $u_i(g_i)$ and the weights p_i. For example, for the additive utility function:

$$u(\mathbf{g}) = \sum_i p_i u_i(g_i),$$

the marginal utilities $u_i(g_i)$ and the weights p_i are obtained by direct interrogation of the decision maker (aggregation methods) as far as it concerns the MAUT approach, and by indirect interrogation of the decision maker (disaggregation methods) as far as it concerns the multiple regression approach. The principal drawback which prevents the closeness of the two approaches is related to the linearity of the models proposed by multiple linear regression. A rather exhaustive bibliography of the methods of the disaggregation of preferences can be found in [32] and [53].

Decision Aid Activity

Concerning the activity of decision aid, Roy [58,60] proposes a methodology of systematic intervention of multicriteria analysis in the decision process. In brief, this methodology comprises four levels:

I) Object of the decision and spirit of recommendation or participation.
II) Analyzing consequences and developing criteria.
III) Modeling comprehensive preferences and operationally aggregating performances.
IV) Investigating and developing the recommendation.

It is important to emphasize that these four levels do not necessarily follow one another in the above mentioned order. The activity of decision aid does not necessarily constitute a sequential process; interactions between the decision maker and the analyst can occur. This general methodology has contributed to the development of several multicriteria methods which have been applied successfully to real cases. Among these methods the well-known are the ELECTRE methods developed by Roy and his collaborators.

Multicriteria Character of Financial Problems and Some Real-World Applications

The operational research techniques were the first to be used in the solution of some financial problems. I. Ekeland [19] wonders

> why finance, rather curiously, has remained so long away from the techniques of operational research (i. e. optimization techniques), except for those concerning portfolio selection models.

According to the same author, the Capital Asset Pricing Model (CAPM) is a static optimization model based on the principle according which, the best portfolio (i. e. optimal portfolio) is the one which maximizes the expected return for a given level of risk, in the period of time considered. For R.W. Ashford et al. [2], the techniques of operational research can be applied to working capital management as well as to the evaluation of investment projects. Among the techniques used for the management of working capital, one could mention:

- inventory control for the management of stocks;
- dynamic programming, linear programming, stochastic programming and visual and interactive techniques of simulation for the management of cash;
- the Markov process and the discriminant analysis for the management of accounts receivable;
- dynamic programming, linear programming, and stochastic programming for the management of short-term debts (current liabilities).

Among the techniques used in the evaluation of investment projects, one could mention the simulation methods [23] and those of mathematical statistics [24] which take into consideration the risk factor. Simulation methods and linear programming (i. e. the LONGER program, [51]) are also used in financial planning (i. e. elaboration of investment and financing plans). Under these circumstances, the solution of financial problems is easy to obtain. It is based on the fact that the problem is well posed, well-formulated regarding the reality involved and on an evaluation criterion (i. e. monocriteria paradigm). But in reality, the modeling of financial problems is based on a different kind of logic. In that case, their solution should take into consideration the following elements (i. e. multicriteria paradigm, cf. [59]):

- multiple criteria;
- conflict situation between the criteria;
- complex evaluation process, subjective and ill-structured;
- introduction of financial decision makers in the evaluation process.

MCDA has already contributed in a significant manner to the solving of several financial problems such as venture capital investment, business failure risk, credit granting, bond rating, country risk, political risk, evaluation of the performance and viability of organizations, choice of investments, financial planning and portfolio management.

The multicriteria character of these financial problems can be easily demonstrated. We will limit here the analysis on the choice of investment projects and portfolio management. International literature could actually provide very important case studies for the rest of the financial problems [36,80,84].

Investment Decision

The choice of investment projects entails an important decision for every firm, public or private, large or small one. In fact, considering its duration, its amount and its irreversible character an investment decision is regarded as a major and strategic one. Therefore, the process of an *investment decision* should be conveniently modeled. If one considers that, in principle, the investment decision process consists of four main stages: perception, formulation, evaluation and choice, the financial theory intervenes only in the stages of evaluation and choice [8]. With its empirical financial criteria (i. e. the payback method, the accounting rate of return) and sophisticated ones, based on discount techniques (i. e. the net present value, the internal rate of return, the index of profitability, the discounted payback method, etc.), the financial theory proposes either a ranking from the better to worst when there are many investment projects in competition or an acceptance or refusal if there is only one investment project. Although the tools of the financial theory should be improved so that they could take into account time, inflation and risk (i. e. analytical methods, simulation methods, games theory, CAPM, etc.), there are still problems concerning the evaluation and selection of investment projects. Among the most important ones, one could mention the reduction of the investment notion in a time series of monetary flows (i. e. inflows, outflows), the choice of the discount rate, the conflicts between financial criteria (i. e. net present value versus internal rate of return), etc. According to the financial theory, the discount rate (sometimes rate of return) plays the role of acceptance or rejection rate (a cut off rate) of an investment project in the case where the criterion of internal rate of return is used. Thus, one can see that the investment decision of a firm depends on one variable only, which is the discount rate. As far as the conflicts between criteria are concerned, one often ascertain that the criteria which are supposed to express the goal of the profitability of projects, could lead to divergent rankings (for example, the net present value and the index of profitability or even the net present value and the internal rate of return). In consequence, the financial approach of investment decision seems limited and unrealistic. It is limited because it remains in the stages of evaluation and choice, and it is unrealistic because it is based only on financial criteria.

MCDA, on the other hand, contributes in a very original way to the investment decision process. Initially, it intervenes in the whole process of investment, from the stages of perception and formulation to the stages evaluation and choice. Concerning the stages of perception and formulation, MCDA contributes to the identification of possible actions (i. e. investment opportunities) and to the definition of a set of potential actions (i. e. possible variants, each variant constituting an investment project in competition with others). This set of projects can be global, fragmented, stable or evolutionary. Then, it is necessary to choose a reference problematic which is well-adapted to the investment decision problem (i. e. choice, sorting, ranking).

- Choice problematic $P.\alpha$: help in choosing the best investment project or in developing a selection procedure for investment projects.
- Sorting problematic $P.\beta$: help in sorting investment projects according to norms or in building an assignment procedure for investment projects.
- Ranking problematic $P.\gamma$: help in ranking the investment projects according to a decreasing preference order or in building an ordering procedure for investment projects.

Concerning the stages of evaluation and choice, MCDA offers a methodological framework much more realistic than the financial theory, by introducing in the study of investment projects both quantitative and qualitative criteria. Criteria such as the urgency of the project, the coherence of the objectives of the projects with those of the general policy of the firm [21], the social and environmental aspects should be taken into consideration in an investment decision. Therefore, MCDA contributes to show the best investment projects according to the problematic chosen, to solve the conflicts between criteria satisfactorily, to set up the relative importance of criteria in the decision making process and to make known the preferences and the investors' system of values. It is very interesting to mention that many authors have already used MCDA methods in the evaluation of investment projects (list non exhaustive): ELECTRE II and ORESTE methods [14]; MAUT methods [21]; multi-objective mathematical programming [5,35,41]; the Analytic Hierarchy Process (AHP) method [38]; PROMETHEE method [55,81].

Finally, in order to examine if the firms apply in reality multiple criteria in their investment decisions, we present the results of the empirical study of Bhaskar and McNamee [6]. The two authors, by studying large United Kingdom companies, have shown that most companies appear to have more than one objective when an investment is being appraised (96%). The most common number of objectives that companies had was eight. Concerning the objectives priority, most companies (77%) had profitability as the primary objective. The next most important objective was company's growth. Other criteria less important than the two above but, which play a role in the investment decisions are the risk, the liquidity, the environment, the age of assets, the flexibility, the depth of skills, etc. With these empirical results an answer has been given to the questions posed in the introduction by the two authors.

Portfolio Management

In the field of *portfolio management* it is possible to cite the pioneering work of H.M. Markowitz [46] who, by developing the optimization model mean-variance (M-V), is the founder of the classical approach of the portfolio management. According to [19], the problem of portfolio choice in the model (M-V) is a multicriteria one, because the investor will try simultaneously to maximize the return and minimize the risk; but determining the acceptance level of risk, one comes back to maximize the return, which is a classical monocriteria problem. After this bicriteria, and even more the monocriteria (i. e. market model, CAPM) portfolio choice consideration, the development of multifactor models has been started where there are more types of risk and not only market risk [57]. Thus, the problem of portfolio selection becomes multidimensional. The necessity of having multidimensional methods (i. e. statistics and econometrics) for the selection of stocks has been presented by specialist researchers in finance [33]. The multidimensional nature of risk in portfolio management has also been demonstrated by specialist researchers in multicriteria analysis. See [76,77] and [10] on the 'Prospect Ranking Vector (PRV)' method. Today an arsenal of multidimensional and multicriteria methods such as factor analysis, goal programming, AHP, ELECTRE, MINORA, ADELAIS, etc. have been already applied in the field of portfolio management [9,25,27,28,29,37,40,47,48,62,82,86].

The multicriteria nature of the problem of portfolio selection is well presented in [37]. The authors study the problem of the international portfolio selection. According to them, the classical optimization model of portfolio selection used in a national context can have even more chance of being sub-optimal in a situation of international diversification. In fact, in an international context, the M-V model does not always constitute a suitable method because, it does not incorporate all the criteria that the portfolio managers and investors use in their stock investment decisions. For such decisions, the authors propose new criteria such as: the return of the last five years on a monthly basis, the standard deviation of the return calculated on the last five years, the total cost of transactions, the country risk (or political risk), the direct available coverage for foreign currencies and the exchange risk. The multicriteria methodology used (i. e. ELECTRE IS, ELECTRE III) has the advantage of offering the portfolio manager a large set of investment opportunities, and also gives him the flexibility of choosing the relative importance of the different criteria during the process of portfolio selection. Finally, the authors believe that the use of an optimization model under constraints changes the nature of the portfolio selection problem because a constraint does not play the same role as a criterion in all decision problems. To show this new direction of research in portfolio management, it is convenient to mention the special issue of the Canadian journal 'L' Actualité Economique}, which is dedicated on the contribution of multicriteria analysis in the study of financial markets [36].

Some Real-World Applications

In this paragraph two applications of MCDA are briefly presented. The first one concerns the evaluation of the venture capital investment and, the second one the evaluation of the business failure risk.

Venture Capital Investment

Venture capital constitutes today an important source of financing for small and medium size firms. It plays, also, an interesting role in the development of the business' spirit. The crucial problem for *venture capital investment* is the choice of evaluation criteria and their aggregation in a global operational model, which will serve as a basis for the rational and automatic selection of viable firms. The earlier evaluation models (i. e. descriptive and statistical) can not explain the investment decisions in venture capital, since the latter relies much more on subjective and qualitative elements than on objectives and quantitative ones [83]. Moreover, the complexity of the evaluation of venture capital investment problem has been mentioned in the evaluation procedures of projects by two French venture capital firms [80].

Study Context

The data sample coming from two French venture capital firms, IDI and SIPAREX, was used as the application object of MCDA. Although these two firms use project evaluation procedures, their problem remains that of the absence of a model able of supporting their decisions in venture capital investment. In fact, the variables used in the evaluation procedure are both financial variables (i. e. profitability ratios, solvency ratios, liquidity ratios, etc.), and qualitative variables (i. e. market trend, information security, quality of management, market niche/position, etc.). But, although there are, in both venture capital firms, techniques for the treatment of financial variables, there is no explicit model for the elaboration and modeling of the qualitative variables. Therefore, it is at this stage of analysis that the evaluation problem becomes complex. Moreover, the complexity of the evaluation of venture capital investment problem is also underlined in other studies [18,26,54,71,83] among others). The role of the venture capitalist goes beyond that of the simple contributor to the funds of the firm.

Multicriteria Method and Results

The multicriteria system MINORA (Multicriteria INteractive Ordinal Regression Analysis) was proposed for the evaluation of firms to the two venture capital firms. It belongs to the fourth category of MCDA methods, which is the approach of the disaggregation of preferences. The MINORA system is both based on the iterative utilization of an ordinal regression method and on an appropriate man-machine dialogue. Its aim is to construct multicriteria decision models which are as consistent as possible with the judgement policy of a decision maker. The decision maker (here the venture capitalist) expresses his judgement policy by ranking some firms, among those he knows well on the basis of previous decisions. The system, then, by the use of the ordinal regression method UTA (UTilitś Additives [31], estimates optimally the additive utility function(s), on multiple criteria, which is (are) as consistent as possible with the decision maker's ranking. The utility function model is estimated iteratively and interactively. It allows, first, the aggregation of all the criteria (i. e. financial and/or qualitative) by giving their relative importance, and second, the automatic and global evaluation of each firm. With the help of the decision makers of the two venture capital firms, two evaluation models were elaborated (one for each venture capital firm). This paper presents only the global model of IDI.

- The evaluation model for IDI

IDI evaluates firms for financing according to twelve criteria. The utility function model was then estimated in the fourth stage of interaction and appeared perfectly consistent with the objectives of IDI. The equation for the global model is the following:

$$
\begin{aligned}
u(\mathbf{g}) &= 0.008u_1(g_1)\\
&+ 0.072u_2(g_2) + 0.006u_3(g_3) + 0.197u_4(g_4)\\
&+ 0.105u_5(g_5) + 0.232u_6(g_6) + 0.009u_7(g_7)\\
&+ 0.094u_8(g_8) + 0.047u_9(g_9) + 0.071u_{10}(g_{10})\\
&+ 0.097u_{11}(g_{11}) + 0.062u_{12}(g_{12}),
\end{aligned}
$$

where the evaluation criteria are the following:

g_1) the sensitivity of sales to the inflation rate;

g_2) the sensitivity of value added to the sales variations;

g_3) the sensitivity of labor productivity (value added per capita) to wage cost increase (wage per capita);

g_4) the supplier credit in days;

g_5) the available net income;

g_6) the quality of management;

g_7) the research and development effort;

g_8) the extent of diversification;

g_9) the market trend;

g_{10}) the market niche/position;

g_{11}) the cash-out method (opportunities for exit);

g_{12}) the world market share.

The model described above is the best adapted to express the preferences, the knowledge and the experiences of the venture capitalist concerning the quality of the firms and their final evaluation. A detailed presentation of the multicriteria method and the results of the application in the two venture capital firms IDI and SIPAREX can be found in [80].

The Business Failure Risk

According to a general definition, failure is the situation that a firm cannot pay lenders, preferred stock shareholders, suppliers, etc., or a bill is overdrawn, or the firm is bankrupt according to law. Today, there is a complete arsenal of evaluation methods for the *business failure risk* [16]. Since the late 1980s, methods close to a qualitative definition of business failure have been developed. These are multicriteria methods which present undeniable advantages in matter of evaluation for the business failure risk [84].

Study Context

The study concerns the evaluation of failure risk of firms financed by a Greek bank of industrial development. This bank finances with stock equity and long term credit the development of Greek firms and contributes to the renovation of industrial and commercial firms on a national and regional level. As in the previous case of the venture capital investment, there is no model able to provide help to the bank credit managers in the financing of firms.

Multicriteria Method and Results

ELECTRE TRI method was proposed for the evaluation of business failure risk, which is particularly suitable for multicriteria sorting problems. It belongs to the third category of MCDA methods, which is the approach of outranking relations [61,75]. From a finite set of actions (i. e. firms) evaluated by quantitative and/or qualitative criteria and from a set of categories previously defined (i. e. reference actions or reference profiles), ELECTRE TRI proposes two different procedures of assignment which allow the classification of all the actions in these categories. In consequence, ELECTRE TRI consists of establishing an outranking relation between the actions to be assigned and the reference profiles. The eventual differences between the two assignment procedures, the pessimistic and the optimistic one, come from the incomparability situations between an action and one or several reference profiles [17,75].

For the case of the evaluation of the business failure risk, three categories of risk were determined by the credit managers of the Greek bank:

C_1) the failed firms (9 in number);
C_2) the risky firms; uncertain category of firms to be studied further (10 in number);
C_3) the healthy firms (20 in number).

These 39 firms were evaluated by seven criteria, five financial ratios and two strategic criteria. The criteria are the following:

x_1) Earnings before interests and taxes/Total assets,
x_2) Net income/Stockholder's equity,
x_3) Total debts/Total assets,
x_4) Financial expenses/Sales,
x_5) Administrative and general expenses/Sales,
x_6) Managers work experience,
x_7) Market niche/position.

From the reference profiles and the thresholds of discrimination (preference model established by the credit managers of the bank), ELECTRE TRI provided good percentages of classification, which were of the order of 82.05% and 89.74% for the optimistic and the pessimistic procedures respectively. The pessimistic procedure gave better results and proved more adaptable to the problem of evaluation of business failure risk (it did not give serious classification errors of the type $C_1 \to C_3$ or $C_3 \to C_1$). For a detailed presentation of the multicriteria method and the results, see [84].

Concerning other financial problems which present a multicriteria character and on which a MCDA method has been applied, it is possible to provide a list of published works (non exhaustive).

- Acquisitions of firms: [68].
- Bankruptcy risk: [1,17,67,78,79].
- Country risk: [7,11,12,49,52,70].
- Evaluation of performance of organizations
  - Insurance: [45].
  - Banks: [43,44,85].
  - Firms: [4,15,30,39,42,66,87,88].
- Financial planning: [20,22,73].
- Venture capital: [50,56,65].

Concluding Remarks

This article has shown the contribution of the MCDA to the solution of some financial decision problems (i. e. venture capital, business failure risk, investment choice, portfolio management, etc.). In the past, all these problems were approached with the use of financial theory in a very narrow framework, that of optimization. Some researchers took advantage of the optimal character of these problems in order to propose operational research techniques (i. e. classical or monocriteria modeling) for their solution. The use of MCDA methods provides many advantages in financial management, among which one could mention the following:

- the possibility of structuring complex evaluation problems;
- the introduction of both quantitative (i. e. financial ratios) and qualitative criteria in the evaluation process;
- the transparency in the evaluation, allowing good argumentation in financial decisions;
- the introduction of sophisticated scientific methods in the field of financial management.

In conclusion, MCDA methods seem to have a promoting future because they offer a highly methodological and realistic framework to decision problems.

See also

▶ Bi-objective Assignment Problem
▶ Competitive Ratio for Portfolio Management
▶ Decision Support Systems with Multiple Criteria
▶ Estimating Data for Multicriteria Decision Making Problems: Optimization Techniques
▶ Financial Optimization
▶ Fuzzy Multi-objective Linear Programming
▶ Multicriteria Sorting Methods
▶ Multi-objective Combinatorial Optimization
▶ Multi-objective Integer Linear Programming
▶ Multi-objective Optimization and Decision Support Systems
▶ Multi-objective Optimization: Interaction of Design and Control
▶ Multi-objective Optimization: Interactive Methods for Preference Value Functions
▶ Multi-objective Optimization: Lagrange Duality
▶ Multi-objective Optimization: Pareto Optimal Solutions, Properties

- ▶ Multiple Objective Programming Support
- ▶ Outranking Methods
- ▶ Portfolio Selection and Multicriteria Analysis
- ▶ Preference Disaggregation
- ▶ Preference Disaggregation Approach: Basic Features, Examples From Financial Decision Making
- ▶ Preference Modeling
- ▶ Robust Optimization
- ▶ Semi-infinite Programming and Applications in Finance

References

1. Andenmatten A (1995) Evaluation du risque de défaillance des emetteurs d'obligations: Une approche par l'aide multicritère à la décision. Press. Polytechniques Univ. Romandes
2. Ashford RW, Berry RH, Dyson RG (1988) Operational research and financial management. Europ J Oper Res 36:143–152
3. Baumol WJ (1959) Economic theory and operations analysis. Prentice-Hall, Englewood Cliffs, NJ
4. Bergeron M, Martel JM, Twarabimenye P (1996) The evaluation of corporate loan applications based on the MCDA. J Euro–Asian Management 2(2):16–46
5. Bhaskar K (1979) A multiple objective approach to capital budgeting. Accounting and Business Res Winter, pp 25–46
6. Bhaskar K, Mcnamee P (1983) Multiple objectives in accounting and finance. J Business Finance and Accounting 10(4):595–621
7. Clei J (1994) La méthodologie d'analyse de la Coface. Banque Stratégie 109:5–6
8. Colasse B (1993) Gestion financière de l'entreprise. Press. Univ. France, Paris
9. Colson G, De Bruyn Ch (1989) An integrated multiobjective portfolio management system. Math Comput Modelling 12(10-11):1359–1381
10. Colson G, Zeleny M (1979) Uncertain prospects ranking and portfolio analysis under the condition of partial information. Math Syst in Economics 44
11. Cook WD, Hebner KJ (1993) A multicriteria approach to country risk evaluation: With an example employing Japanese data. Internat Rev Economics and Finance 2(4):327–438
12. Cosset JC, Siskos Y, Zopounidis C (1995) Evaluating country risk: A decision support approach. Global Finance J 3(1):79–95
13. Cyert RM, March JG (1963) A behavioral theory of the firm. Prentice-Hall, Englewood Cliffs, NJ
14. Danila N (1980) Méthodologie d'aide à la décision dans le cas d'investissements fort dépendants. Thèse de Doctorat de 3e Cycle, UER Sci des Organisations, Univ Paris-Dauphine
15. Diakoulaki D, Mavrotas G, Papagyannakis L (1992) A multicriteria approach for evaluating the performance of industrial firms. OMEGA Internat J Management Sci 20(4):467–474
16. Dimitras AI, Zanakis SH, Zopounidis C (1996) A survey of business failures with an emphasis on prediction methods and industrial applications. Europ J Oper Res 90:487–513
17. Dimitras AI, Zopounidis C, Hurson CH (1995) A multicriteria decision aid method for the assessment of business failure risk. Found Computing and Decision Sci 20(2):99–112
18. Dixon R (1991) Venture capitalists and the appraisal of investments. OMEGA Internat J Management Sci 19(5):333–344
19. Ekeland I (1993) Finance: Un nouveau domaine des mathématiques appliquées. Revue Franc de Gestion, pp 44–48
20. Eom HB, Lee SM, Snyder ChA, Ford FN (1987/8) A multiple criteria decision support system for global financial planning. J Management Information Systems 4(3):94–113
21. Evrard Y, Zisswiller R (1982) Une analyse des décisions d'investissement fondée sur les modèles de choix multiattributs. Finance 3(1):51–68
22. Goedhart MH, Spronk J (1995) Financial planning with fractional goals. Europ J Oper Res 82:111–124
23. Hertz DB (1964) Risk analysis in capital investment. Harvard Business Rev 42:95–106
24. Hillier FS (1963) The derivation of probabilistic information for the evaluation of risky investments. Managem Sci 9:443–457
25. Ho PC, Paulson AS (1980) Portfolio selection via factor analysis. J Portfolio Management, pp 27–30
26. Hoban JP (1976) Characteristics of venture capital investments. Unpublished Doctoral Diss, Univ Utah
27. Hui TK, Kwan E (1994) Internat. portfolio diversification: A factor analysis approach. OMEGA Internat J Management Sci 22(3):263–267
28. Hurson CH, Zopounidis C (1995) On the use of multicriteria decision aid methods to portfolio selection. J Euro–Asian Management 1(2):69–94
29. Hurson CH, Zopounidis C (1997) Gestion de portefeuille et analyse multicritère. Economica, Paris
30. Jablonsky J (1993) Multicriteria evaluation of clients in financial houses. Central Europ J Oper Res and Economics 3(2):257–264
31. Jacquet-Lagrèze E, Siskos J (1982) Assessing a set of additive utility functions for multicriteria decision making: The UTA method. Europ J Oper Res 10:151–164
32. Jacquet-Lagrèze E, Siskos J (1983) Méthodes de décision multicritère. Ed. Hommes et Techniques
33. Jacquillat B (1972) Les modèles d'évaluation et de sélection des valeurs mobilières: Panorama des recherches américaines. Analyse Financière 11(4):68–88
34. Keeney RL, Raiffa H (1976) Decisions with multiple objectives: Preferences and value tradeoffs. Wiley, New York

35. Khorramshahgol R, Okoruwa AA (1994) A goal programming approach to investment decisions: A case study of fund allocation among different shopping malls. Europ J Oper Res 73:17–22
36. Khoury NT, Martel JM (1993) Nouvelles orientations dans l'étude des marchés financiers: Asymétrie d'information et analyse multicritère. L'Actualité Economique, Revue d'Analyse Economique 69(1):5–7
37. Khoury NT, Martel JM, Veilleux M (1993) Méthode multicritère de sélection de portefeuilles indiciels internationaux. L'Actualité Economique, Revue d'Analyse Economique 69(1):171–190
38. Kivijärvi H, Tuominen M (1992) A decision support system for semistructured strategic decisions: A multi-tool method for evaluating intangible investments. Revue des Systèmes de Décision 1:353–376
39. Lee H, Kwak W, Han I (1995) Developing a business performance evaluation system: An analytic hierarchical model. The Engineering Economist 30(4):343–357
40. Lee SM, Chesser DL (1980) Goal programming for portfolio selection. J Portfolio Management, pp 22–26
41. Lin WT (1978) Multiple objective budgeting models: A simulation. Accounting Rev LIII(1):61–76
42. Mareschal B, Brans JP (1993) BANKADVISER: Un système interactif multicritère pour l'évaluation financière des entreprises à l'aide des méthodes PROMETHEE. L'Actualité Economique, Revue d'Analyse Economique 69(1):191–205
43. Mareschal B, Mertens D (1990) Evaluation financière par la méthode multicritère GAIA: Application au secteur bancaire belge. Revue de la Banque 6:317–329
44. Mareschal B, Mertens D (1992) BANKS: A multicriteria, PROMETHEE-based, decision support system for the evaluation of the international banking sector. Revue des Systèmes de Décision 1(2-3):175–189
45. Mareschal B, Mertens D (1993) Evaluation multicritère par la méthode multicritère GAIA: Application au secteur de l'assurance en Belgique. L'Actualité Economique, Revue d'Analyse Economique 69(1):206–228
46. Markowitz H (1952) Portfolio selection. J Finance, pp 77–91
47. Martel JM, Khoury NT, Bergeron M (1988) An application of a multicriteria approach to portfolio comparisons. J Oper Res Soc 39(7):617–628
48. Martel JM, Khoury NT, M'zali B (1991) Comparaison performance-taille des fonds mutuels par une analyse multicritère. L'Actualité Economique, Revue d'Analyse Economique 67(3):306–324
49. Mondt K, Despontin M (1986) Evaluation of country risk using multicriteria analysis. Paper presented at the EURO VIII Conf
50. Muzyka D, Birley S, Leleux B (1996) Trade-offs in the investment decisions of Europ. venture capitalists. J Business Venturing 11:273–287
51. Myers SC, Pogue GA (1974) A programming approach to corporate financial management. J Finance 29:579–599
52. Oral M, Kettani O, Cosset JC, Daouas M (1992) An estimation model for country risk rating. Internat J Forecasting 8:583–593
53. Pardalos PM, Siskos Y, Zopounidis C (1995) Advances in multicriteria analysis. Kluwer, Dordrecht
54. Poindexter JB (1976) The efficiency of financial markets: The venture capital case. Unpublished Doctoral Diss New York Univ
55. Ribarovic Z, Mladineo N (1987) Application of multicriterional analysis to the ranking and evaluation of the investment programmes in the ready mixed concrete industry. Engin Costs and Production Economics 12:367–374
56. Riquelme H, Rickards T (1992) Hybrid conjoint analysis: An estimation probe in new venture decisions. J Business Venturing 7:505–518
57. Ross SA (1976) The arbitrage theory of capital asset pricing. J Econom Theory, pp 343–362
58. Roy B (1985) Méthodologie multicritère d'aide à la décision. Economica, Paris
59. Roy B (1988) Des critères multiples en recherche opérationnelle: Pourquoi? In: Rand GK (ed) Operational Research '87. Elsevier, Amsterdam, pp 829–842
60. Roy B (1996) Multicriteria methodology for decision aiding. Kluwer, Dordrecht
61. Roy B, Bouyssou D (1993) Aide multicritère à la décision: Méthodes et cas. Economica, Paris
62. Saaty TL, Rogers PC, Pell R (1980) Portfolio selection through hierarchies. J Portfolio Management Spring:16–21
63. Schärlig A (1996) Pratiquer ELECTRE et PROMETHEE. Press. Polytechniques Univ. Romandes
64. Simon HA (1957) Models of man. Wiley, New York
65. Siskos J, Zopounidis C (1987) The evaluation criteria of the venture capital investment activity: An interactive assessment. Europ J Oper Res 31:304–313
66. Siskos Y, Zopounidis C, Pouliezos A (1994) An integrated DSS for financing firms by an industrial development bank in Greece. Decision Support Systems 12:151–168
67. Slowinski R, Zopounidis C (1995) Application of the rough set approach to evaluation of bankruptcy risk. Internat J Intelligent Systems in Accounting, Finance, and Management 4:27–41
68. Slowinski R, Zopounidis C, Dimitras AI (1997) Prediction of company acquisition in Greece by means of the rough set approach. Europ J Oper Res 100:1–15
69. Spronk J (1981) Interactive multiple goal programming application to financial planning. Martinus Nijhoff, Boston
70. Tang JCS, Espinal CG (1989) A model to assess country risk. OMEGA Internat J Management Sci 17(4):363–367
71. Tyebjee TT, Bruno AV (1984) A model of venture capitalist investment activity. Managem Sci 30(9):1051–1066
72. Vincke Ph (1992) Multicriteria decision-aid. Wiley, New York
73. Vinso JD (1982) Financial planning for the multinational corporation with multiple goals. J Internat Business Stud, pp 43–58

74. Williamson OE (1964) The economics of discretionary behavior: Managerial objectives in a theory of the firm. Prentice-Hall, Englewood Cliffs, NJ
75. Yu W (1992) Aide multicritère à la décision dans le cadre de la problématique du tri: Concepts, méthodes et applications. Thèse de Doctorat, Univ Paris-Dauphine
76. Zeleny M (1977) Multidimensional measure of risk: The prospect ranking vector. In: Zionts S (ed) Multiple Criteria Problem Solving. Springer, Berlin, pp 529–548
77. Zeleny M (1982) Multiple criteria decision making. McGraw-Hill, New York
78. Zollinger M (1982) L'analyse multicritère et le risque de crédit aux entreprises. Revue Franc de Gestion Jan.-Fév.:56–66
79. Zopounidis C (1987) A multicriteria decision-making methodology for the evaluation of the risk of failure and an application. Found Control Eng 12(1):45–67
80. Zopounidis C (1990) La gestion du capital-risque. Economica, Paris
81. Zopounidis C (1993) A multicriteria methodology for the evaluation and ranking of investment projects. Bull Greek Banks Assoc 10(39-40):22–28. (In Greek)
82. Zopounidis C (1993) On the use of the MINORA multicriteria decision aiding system to portfolio selection and management. J Inform Sci and Techn 2(2):150–156
83. Zopounidis C (1994) Venture capital modelling: Evaluation criteria for the appraisal investments. The Financier: ACMT 1(2):54–64
84. Zopounidis C (1995) Evaluation du risque de défaillance de l'entreprise: Méthodes et cas d'application. Economica, Paris
85. Zopounidis C, Despotis DK, Stavropoulou E (1995) Multiattribute evaluation of Greek banking performance. Applied Stochastic Models and Data Analysis 11(1):97–107
86. Zopounidis C, Godefroid M, Hurson Ch (1995) Designing a multicriteria decision support system for portfolio selection and management. In: Janssen J, Skiadas CH, Zopounidis C (eds) Advances in Stochastic Modelling and Data Analysis. Kluwer, Dordrecht, pp 261–292
87. Zopounidis C, Matsatsinis NF, Doumpos M (1996) Developing a multicriteria knowledge-based decision support system for the assessment of corporate performance and viability: The FINEVA system. Fuzzy Economic Rev 1(2):35–53
88. Zopounidis C, Pouliezos A, Yannacopoulos D (1992) Designing a DSS for the assessment of company performance and viability. Comput Sci in Economics and Management 5:41–56

Financial Equilibrium

Anna Nagurney
University Massachusetts, Amherst, USA

MSC2000: 91B50

Article Outline

Keywords
A Multi-Sector, Multi-Instrument Financial Equilibrium Model
Optimality Conditions
Economic System Conditions
See also
References

Keywords

Efficient frontier; Risk-free asset; Market portfolio; Option; Perfect competition; Portfolio optimization; Variational inequality formulation

Finance is concerned with the study of capital flows over space and time. The theory of financial economics is a combination of many different theories among which the theories of finance and economics, mathematical programming, and utility theory are credited with the biggest contributions.

The current state of modern financial economic theory is based upon the fundamental contributions of economists in the decade of the 1950s. Here we review some of the major developments. For a more complete historical breakdown, see [32].

The first major breakthrough was by K. Arrow and G. Debreu, who, in a series of publications (cf. [1,2,4,12,13]), introduced an important extension to the existing economic theory. Their contributions brought competitive equilibrium theory to a new level and allowed for the development of modern economic and finance theory. Specifically, Arrow and Debreu applied the techniques of convexity and fixed point theory to a model that followed the neoclassical economic foundations of: market clearing, uncertainty, and individual rationality and then they derived new fundamental economic properties from these models (e. g., [3,14]).

F. Modigliani and M. Miller [28], in turn, showed that the capital structure of a firm, that is, the financial framework of the firm, usually measured by the debt to equity ratio, does not affect the value of a firm. In their work, for the first time, the idea of financial arbitrage was used by stating that any investor can use risk-less arbitrage in order to avoid the financial structure of a firm. Their work serves as the base for most of the research on capital structure.

The other theoretical breakthrough was by H.M. Markowitz in 1952, the founder of modern portfolio theory. Markowitz [25] proposed that one of the principal objectives of investors, in addition to the maximization of the returns of their portfolios, is to diversify away as much risk as possible. He claimed that investors choose assets in a manner so that the risk of their portfolio matches their risk preferences. He suggested that individuals who cannot bear risk will invest in assets with low risk, whereas people more comfortable with risk will accept investments of higher risk. His work suggested that the trade-off between risk and return is distinct for each investor; however, the preferences of all people lie upon a fictitious curve which is usually called the *'frontier of efficient portfolios'*. Along this curve lie all the diversified portfolios that have the highest return for a given risk, or the lowest risk for a given return. Markowitz's model was based on mean-variance portfolio selection, where the average and the variability of portfolio returns were determined in terms of the mean and covariance of the corresponding investments.

Many versions and extensions of Markowitz's model have appeared in the literature (cf. [19], and the references therein). The first important simplification of Markowitz's model was suggested by W.F. Sharpe [35], through a model known as the *diagonal model*, in which 'the individual covariances between all securities are assumed to be zero'. According to this model, the variance-covariance matrix has zeros in all positions other than the diagonal.

The most significant extension of the models by Markowitz [25] and Sharpe [35], was the *Capital Asset Pricing Model* (CAPM), which was based on the work of Sharpe [36], J. Lintner [24], and J. Mossin [29]. In this model the concept of a risk-free asset and market portfolio were introduced. A *risk-free asset* is an asset with a positive expected rate of return and a zero standard deviation. A *market portfolio*, on the other hand, is a portfolio on the efficient frontier of the Markowitz model which is considered to be desirable by all investors. The CAPM assumes that all investors will select a portfolio that will be a linear combination of the risk-free asset and the market portfolio, and, hence, the equilibrium prices of all assets can be expressed as a linear combination of the risk-free price and the price of the market portfolio. Since some of the assumptions governing the CAPM were not realistic (such as the absence of transaction costs), the model was extended and improved several times in the years that followed. It is, nevertheless, one of the major breakthroughs in modern economic and finance theory and forms the basis for most of the financial models.

Most of the major extensions of the CAPM occurred in the decade of the seventies, where a series of papers either relaxed some of its assumptions, or derived empirical results by applying it to a series of problems. Among the most significant contributions of that time were: the extension to a multiperiod economy by R.C. Merton [27] and the consumption CAPM by D.T. Breeden [6] (which, however, failed empirically due to the difficulty in observing and computing consumption).

The dissatisfaction with the empirical tests of the CAPM led to more advanced models, such as the *Arbitrage Pricing Theory* (APT) by S.A. Ross [34]. The APT's main contribution was the inclusion of multiple risk factors and the generalization of the CAPM, which was considered to be a special case of APT with only a single risk factor. In particular, Ross assumed that the rate of return of every security can be expressed as a linear combination of some 'basic' risk factors.

Another major development in modern financial economic theory was the derivation of an accurate option pricing model by F. Black and M. Scholes [5], which revolutionized the pricing of financial instruments and the entire financial industry. Note that an *option* is, in general, the right to trade an asset for a preagreed amount of capital. If the right is not exercised after a predetermined period of time, the option expires and the holder loses the money paid for holding that right. A major part of the subsequent literature focused on different approaches to, simplifications of, and variations of the *Black–Scholes Model* (BSM). A significant simplification of the BSM was done by J.C. Cox, Ross, and M. Rubinstein [11] (see [27]).

Furthermore, the *mean-variance portfolio analysis* that was introduced and mathematically formulated by Markowitz [25,26] and later simplified by the diagonal model of Sharpe [35], was further extended by G.A. Pogue [33] and J.C. Francis [18], with the introduction of variance-covariance matrices for both assets and liabilities, applied to the asset-liability management of banks.

Most of the aforementioned models and theories were subsequently extended and improved. The APT of Ross was refined by G. Chamberlain [7] and G. Connor [8], and the Black–Scholes model was further explored and significantly generalized (see, e. g., [10,15,17]).

The majority of the literature in financial economics has been based on the assumption that investors cannot affect the prices at which they buy or sell. Each investor is considered to be an isolated case, who tries to maximize his utility function, subject to the prices that the market provides him. All the participants in the economy, be they buyers or sellers, have as a goal the maximization of their profits and the minimization of their losses. The prices are derived through the market where investors constantly buy and sell commodities. The analysis of market equilibrium tries to determine the prices at which different products will be bought and sold, and also the amount of each product that each participant in the economy will hold in an equilibrium state.

Market equilibrium analysis has its roots in the last half of the nineteenth century. The work of H. Gossen [21], W. Jevons [23], and L. Walras [39] initiated the analysis of equilibrium theory. Subsequently, in the 1930s the study of market equilibrium became more formal and solid. The work of A. Wald [37,38] and J.R. Hicks [22] provided, for the first time, proofs of different qualitative properties of the equilibrium, along with a detailed study of the conditions under which an equilibrium could be modeled and derived. Furthermore, the work of Arrow [1] and G. Debreu [12] started a new era in equilibrium analysis by bringing uncertainty into equilibrium theory, which led to the current status of market equilibrium theory.

The basic assumption that governs most of the existing models that address the theory of market equilibrium is that of *perfect competition*. Perfect competition prohibits any participant in the economy (buyer or seller) from having control over the prices of different products or over the actions of other participants. The price of a product is considered to be a variable, the value of which is determined by the combined actions of all the buyers and sellers. Buyers are, hence, *'price takers'*, in that they modify their holdings of a product according to the price, ignoring the effects that their behavior may have on that price. Moreover, perfect competition assumes that all participants in the economy have perfect information about the products available, the current price, and the bids of a specific product. Furthermore, the number of the participants in the economy is assumed to be large enough so that the market activity regarding a specific product will be small compared to the transactions in the overall market.

For definiteness, we present a financial equilibrium model due to A. Nagurney [30] (see, also, [32], and the references therein). The model relaxes the CAPM assumptions of homogeneous expectations (cf. [24,29,36]), without imposing restrictions as to the nature of different sectors (e. g., [20]).

The mathematical framework that is utilized to develop the multi-sector, multi-instrument financial equilibrium model is finite-dimensional variational inequality theory. The methodology of finite-dimensional variational inequalities was first suggested for the modeling, analysis, and computation of multi-sector, multi-instrument financial equilibrium problems by Nagurney, J. Dong, and M. Hughes [31] and was further explored by Nagurney [30]. For complete references, qualitative results, as well as a plethora of financial equilibrium models and computational approaches, see [32].

A Multi-Sector, Multi-Instrument Financial Equilibrium Model

Consider a single country economy with multiple instruments and with multiple sectors. We let i denote a typical instrument, with the total number of instruments available in the economy, denoted by I. We let j denote a typical sector in the economy, with the number of sectors denoted by J.

Let r_i denote the (nonnegative) price of instrument i, and group the prices of all the instruments into the column vector $r \in \mathbf{R}^I_+$. Denote the volume of instrument i that sector j holds as an asset, by X^j_i, and group the (nonnegative) assets in the portfolio of sector j into the column vector $X^j \in \mathbf{R}^I_+$. Further, group the assets of all sectors in the economy into the column vector $X \in \mathbf{R}^{JI}_+$. Similarly, denote the volume of instrument i that sector j holds as a liability, by Y^j_i, and group the (nonnegative) liabilities in the portfolio of sector j into the column vector $Y^j \in \mathbf{R}^I_+$. Finally, group the liabilities of all sectors in the economy into the column vector $Y \in \mathbf{R}^{JI}_+$.

Assume that the total volume of each balance sheet side of each sector is exogenous. Recall that a *balance sheet* is a financial report that demonstrates the status of a company's assets, liabilities, and the owner's equity at a specific point of time. The left-hand side of a balance sheet contains the assets that a sector holds at a particular point of time, whereas the right-hand side accommodates the liabilities and owner's equity held by that sector at the same point of time. According to accounting principles, the sum of all assets is equal to the sum of all the liabilities and the owner's equity. Moreover, we assume that the sectors under consideration act in a perfectly competitive environment.

Since each sector's expectations are formed by reference to current market activity, a sector's expected utility maximization can be written in terms of optimizing the current portfolio. Sectors may trade, issue, or liquidate holdings in order to optimize their portfolio compositions.

We assume that each sector j tries to maximize his utility function, which we denote as $U^j(X^j, Y^j, r)$. We also assume that the utility function of every sector is concave, continuous, and twice continuously differentiable. Furthermore, the accounts of each sector must balance. We denote the total financial volume held by sector j by S^j. Therefore, the optimization problem that each sector j faces is given by:

$$\begin{cases} \max & U^j(X^j, Y^j, r) \\ \text{s.t.} & \sum_{i=1}^{I} X_i^j = S^j, \\ & \sum_{i=1}^{I} Y_i^j = S^j, \\ & X_i^j \geq 0,\ Y_i^j \geq 0, \\ & \quad i = 1, \ldots, I, \end{cases}$$

where the price vector r is an exogenous vector in the optimization problem of every sector j; $j = 1, \ldots, J$.

We now discuss the feasible set of the sectors. For each sector j; $j = 1, \ldots, J$, we let

$$\widetilde{X}^j \equiv \left\{ X^j \in \mathbf{R}^I_+ : \ \sum_{i=1}^{I} X_i^j = S^j \right\}$$

denote the constraint set of his assets. Similarly, we let

$$\widetilde{Y}^j \equiv \left\{ Y^j \in \mathbf{R}^I_+ : \ \sum_{i=1}^{I} Y_i^j = S^j \right\}$$

denote the constraint set for his liabilities. Then, the feasible set for a sector j is a Cartesian product, denoted by κ^j, where

$$\kappa^j \equiv \{\widetilde{X}^j \times \widetilde{Y}^j\}.$$

Let $\widetilde{X}$ denote the feasible set for the assets of all the sectors, where:

$$\widetilde{X} \equiv \widetilde{X}^1 \times \ldots \times \widetilde{X}^j \times \ldots \times \widetilde{X}^J.$$

Similarly, for the liabilities, let $\widetilde{Y}$ denote the feasible set of the liabilities of all the sectors, that is,

$$\widetilde{Y} \equiv \widetilde{Y}^1 \times \ldots \times \widetilde{Y}^j \times \ldots \times \widetilde{Y}^J.$$

Also, define $\kappa \equiv \{\widetilde{X} \times \widetilde{Y}\}$.

We now present the optimality conditions for a sector's utility maximization problem, given above. We then give the economic conditions determining the instrument prices (in equilibrium).

Optimality Conditions

The necessary and sufficient conditions for an optimal portfolio for sector j are that the vector of assets and liabilities, $(X^{j*}, Y^{j*}) \in \kappa^j$, satisfies the following system of equalities and inequalities: For each instrument i, $i = 1, \ldots, I$, we must have the following *Kuhn–Tucker conditions* being satisfied, at an equilibrium price vector r^*:

$$-\frac{\partial U^j(X^{j*}, Y^{j*}, r^*)}{\partial X_i^j} - \mu_j^1 \geq 0,$$

$$-\frac{\partial U^j(X^{j*}, Y^{j*}, r^*)}{\partial Y_i^j} - \mu_j^2 \geq 0,$$

$$X_i^{j*}\left(-\frac{\partial U^j(X^{j*}, Y^{j*}, r^*)}{\partial X_i^j} - \mu_j^1\right) = 0,$$

$$Y_i^{j*}\left(-\frac{\partial U^j(X^{j*}, Y^{j*}, r^*)}{\partial Y_i^j} - \mu_j^2\right) = 0,$$

where μ_j^1, μ_j^2 are the Lagrange multipliers associated with the constraints. Obviously, a similar set of equalities and inequalities holds for every other sector in the single country economy.

Economic System Conditions

Moreover, the economic system conditions that ensure market clearance at a positive instrument price (and a possible excess supply of the instrument at a zero price) are: For each instrument i, $i = 1, \dots, I$, we must have that:

$$\sum_{j=1}^{J}(X_i^{j*} - Y_i^{j*}) \begin{cases} = 0 & \text{if } r_i^* > 0, \\ \geq 0 & \text{if } r_i^* = 0. \end{cases}$$

This system of equalities and inequalities states that if the price of a financial instrument is positive, then the market must clear for that instrument and if the price is zero, then either there is an excess supply of that instrument in the economy or the market clears.

Let K be the feasible set for all the asset and liability holdings of all the sectors, and all the prices of all the instruments where $K \equiv \{\kappa \times \mathbf{R}_+^I\}$.

Combining the above, we present the following definition of equilibrium.

Definition 1 (*financial equilibrium*) A vector $(X^*, Y^*, r^*) \in K$ is an equilibrium of the single country, multi-sector, multi-instrument financial model if and only if it satisfies the system of equalities and inequalities above, for all sectors j, $j = 1, \dots, J$, and for all instruments i, $i = 1, \dots, I$, simultaneously.

The necessary and sufficient conditions for optimal portfolios, along with the economic conditions for the instrument prices, are now utilized in obtaining the variational inequality formulation of the financial equilibrium conditions.

Theorem 2 (***variational inequality formulation***) *A vector of assets and liabilities of the sectors, and instrument prices, $(X^*, Y^*, r^*) \in K$, is a financial equilibrium if and only if it satisfies the variational inequality problem:*

$$\sum_{j=1}^{J}\sum_{i=1}^{I}\left[-\frac{\partial U^j(X^{j*}, Y^{j*}, r^*)}{\partial X_i^j}\right] \times [X_i^j - X_i^{j*}]$$
$$+ \sum_{j=1}^{J}\sum_{i=1}^{I}\left[-\frac{\partial U^j(X^{j*}, Y^{j*}, r^*)}{\partial Y_i^j}\right] \times [Y_i^j - Y_i^{j*}]$$
$$+ \sum_{i=1}^{I}\sum_{j=1}^{J}[X_i^{j*} - Y_i^{j*}] \times [r_i - r_i^*] \geq 0,$$
$$\forall (X, Y, r) \in K.$$

We now put the variational inequality into standard form. We first define the J-dimensional column vector U with components: $\{U^1, \dots, U^J\}$ and let $\nabla_X U$ denote the JI-dimensional vector with components: $\{\nabla_{X^1} U^1, \dots, \nabla_{X^J} U^J\}$ with $\nabla_{X^j} U^j$ denoting the gradient of U^j with respect to the vector X^j. The expression $\nabla_Y U$ is defined accordingly. We let $n = 2JI + I$. We define the n-dimensional column vector $x \equiv (X, Y, r) \in K$, and the n-dimensional column vector $F(x)$ with components:

$$F(x) = \begin{pmatrix} F_1(x) \\ \vdots \\ F_b(x) \\ \vdots \\ F_n(x) \end{pmatrix} = \begin{pmatrix} -\nabla_X U(X, Y, r) \\ -\nabla_Y U(X, Y, r) \\ \sum_{j=1}^{J}(X_1^j - Y_1^j) \\ \vdots \\ \sum_{j=1}^{J}(X_I^j - Y_I^j) \end{pmatrix}_{n \times 1} .$$

Consequently, the variational inequality may be rewritten as:

- Determine $x^* \in K$ satisfying:

$$\left\langle F(x^*)^\top, x - x^* \right\rangle \geq 0, \qquad \forall x \in K.$$

Other financial equilibrium models, including models with hedging instruments such as futures and options, as well as, international financial equilibrium models can be found in [32], and the references therein.

See also

- ► Equilibrium Networks
- ► Generalized Monotonicity: Applications to Variational Inequalities and Equilibrium Problems
- ► Oligopolistic Market Equilibrium
- ► Spatial Price Equilibrium
- ► Traffic Network Equilibrium
- ► Walrasian Price Equilibrium

References

1. Arrow KJ (1951) An extension of the basic theorems of classical welfare economics. Econometrica 51:1305–1323
2. Arrow KJ (1953) Les rôle des valeurs boursières pour la répartition la meilleure des risques. Econométrie Centre Nat Réch Sci, pp 41–48
3. Arrow KJ (1969) Collected Papers of Kenneth Arrow. Belknap Press
4. Arrow KJ, Debreu G (1954) Existence of an equilibrium for a competitive economy. Econometrica 22:265–290
5. Black F, Scholes M (1973) The pricing of options and corporate liabilities. J Political Economy 3:637–654

6. Breeden DT (1978) An intertemporal asset pricing model with stochastic consumption and investment opportunities. J Financial Economics 7:265–296
7. Chamberlain G (1983) Funds, factors and diversification in arbitrage pricing models. Econometrica 51:1305–1323
8. Connor G (1984) A unified beta pricing theory. J Econom Theory 34:13–31
9. Cox JC, Huang C (1987) Option pricing theory and its applications. In: Constantinides G, Bhattacharya S (eds) Frontiers of Financial Theory. Rowman & Littlefield
10. Cox JC, Ingersoll J, Ross SA (1985) An intertemporal general equilibrium model of asset prices. Econometrica 53:363–384
11. Cox JC, Ross SA, Rubinstein M (1979) Option pricing: A simplified approach. J Financial Economics 7:229–263
12. Debreu G (1951) The coefficient of resource utilization. Econometrica 19:273–292
13. Debreu G (1959) Theory of value. Yale Univ. Press
14. Debreu G (1970) Economies with a finite set of equilibria. Econometrica 38:387–392
15. Duffie D (1986) Stochastic equilibria: Existence, spanning number and the 'No expected financial gain from trade' hypothesis. Econometrica 54:1161–1184
16. Duffie D (1988) Security markets. Stochastic models. Acad. Press, New York
17. Duffie D, Huang C (1985) Implementing Arrow-Debreu equilibria by continuous trading of few long-lived securities. Econometrica 53:1337–1356
18. Francis JC (1978) Portfolio analysis of asset and liability management in small-medium-and large-sized banks. J Monetary Economics 3:112–134
19. Francis JC, Archer SH (1979) Portfolio analysis. Prentice-Hall, Englewood Cliffs, NJ
20. Gonedes NJ (1976) Capital market equilibrium for a class of heterogeneous expectations in a two-parameter world. J Finance 31:1–15
21. Gossen H (1854) Entwicklung der Gesetze des Menschlichen Verkehrs. Prager
22. Hicks JR (1939) Value and capital. Clarendon Press, Oxford
23. Jevons W (1871) The theory of political economy. MacMillan, New York
24. Lintner J (1965) The valuation of risk assets and the selection of risky investments in stock portfolios and capital budgets. Rev Economic Stud 47:13–37
25. Markowitz HM (1952) Portfolio selection. J Finance 7:77–91
26. Markowitz HM (1959) Portfolio selection: Efficient diversification of investments. Wiley, New York
27. Merton RC (1973) An intertemporal capital asset pricing model. Econometrica 41:867–887
28. Modigliani F, Miller M (1958) The cost of capital, corporate cinance and the theory of corporation finance. Amer Economic Rev 48:261–297
29. Mossin J (1966) Equilibrium in a capital asset market. Econometrica 34:768–783
30. Nagurney A (1994) Variational inequalities in the analysis and computation of multi-sector, multi-instrument financial equilibria. J Econom Dynam Control 18:161–184
31. Nagurney A, Dong J, Hughes M (1992) Formulation and computation of general financial equilibrium. Optim 26:339–354
32. Nagurney A, Siokos S (1997) Financial networks: Statics and dynamics. Springer, Berlin
33. Pogue GA (1970) An extension of the Markowitz portfolio selection model to include transaction costs, short sales, leverage policies, and taxes. J Finance 25:1005–1027
34. Ross SA (1976) Arbitrage theory of capital asset pricing. J Econom Theory 13:341–360
35. Sharpe WF (1963) A simplified model for portfolio analysis. Managem Sci 9:277–293
36. Sharpe WF (1964) Capital asset prices: A theory of market equilibrium under conditions of risk. J Finance 19:425–443
37. Wald A (1935) Über die eindeutige positive Losbarkeit der neuen Productionsgleichungen. Ergebn Math Kolloq 6:12–20
38. Wald A (1936) Über die productionsgleichungen der Ökonomischen wertlehre. Ergebn Math Kolloq 7:1–6
39. Walras L (1874) Eléments d'economie politique pure. Guillaumin

Financial Optimization

John M. Mulvey[1], Bala Shetty[2]

[1] Department Operations Research & Financial Engineering, Princeton University, Princeton, USA

[2] Department Information & Operations Management, Texas A&M University, College Station, USA

MSC2000: 91B28

Article Outline

Keywords
Single-Period Models
Multiperiod Models
  Parameters
  Decision Variables
  Model SP
Scenario Generation
Solution Techniques
  Direct Solvers
  Decomposition Algorithms
Conclusions and Future Directions
See also
References

Keywords

Stochastic programming; Nonlinear optimization; Network programming

There is great need for an integrative approach to financial analysis and planning. The globalization of financial markets and the introduction of complex products such as exotic derivatives have increased volatility and risks. Strides in computers and information technology has eliminated any delays between the occurrence of an event and the impact on the markets — within the home country and internationally. The domain of *financial planning* provides a rich source of applications for optimization models and related tools. Such tools as simulation, estimation, stochastic processes, decision support, and artificial intelligence have become indispensable in several domains of financial operations [36]. With the continued growth of complex financial instruments and an increased acceptance of operations research tools by practitioners, optimization models are positioned to play a significant role in financial planning. There is a wealth of literature available regarding the role of optimization models in financial planning. See [12,16,23,32,35,37,38].

The primary purpose of this article is to present an overview of an integrative optimization-based financial planning model. In financial applications, the planner must provide recommendations from among a large of number of alternatives in which there is considerable uncertainty. The financial planner must therefore model the decisional environment as well as the stochastic elements in a dynamic fashion. The model presented here encompasses several popular approaches to the problem of investment strategies, including stochastic programs and dynamic stochastic control [4]. The financial planning model results in large stochastic optimization problems and efficient algorithms are now available for solving these nonlinear programs. A brief review of the various algorithms is also presented.

Single-Period Models

The most widely used methods for portfolio selection are based on the mean/variance approach [20]. Mean-variance optimization is a mathematical tool that creates a portfolio of assets with the maximum expected return for a given level of risk or with the minimum risk for a given expected return. Over the years, a number of researchers have extended and refined the original model to include transactions costs, trading size and turnover constraints and other practical requirements [30]. Several researchers have provided efficient procedures for estimating the variance/covariance matrix of returns required by the model, based on factor, index or scenario analysis [10].

While mean-variance analysis provides a powerful framework for asset allocation, it suffers from several limitations. The Markowitz model treats expected returns, standard deviations, and correlations as population parameters. These population parameters are not available, and therefore statistical estimates are used. The estimation errors thus introduced can distort the optimization results and could result in major errors in asset allocation.

Single-period models cannot capture long-term investment goals. They do not have the ability to consider opportunity costs that should influence decisions on strategic placement of funds; investment opportunities with maturities exceeding a single period cannot be included; neither can the impact of anticipated exogenous supply/demand for funds be properly assessed [21]. Single-period models tend to produce high portfolio turnovers and opportunistic asset trades. They cannot accurately account for the effect of transaction costs. Purchases of asset categories with high transaction costs are disfavored unless they promise high immediate returns. Multiperiod models, properly formulated, can overcome many of these limitations. This is the focus of our discussion in the next section.

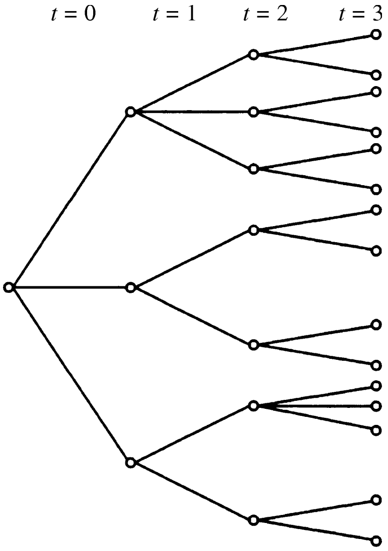
Multiperiod Models

We address financial planning over long horizons via multistage stochastic programming. The stochastic program brings together all major financial-related decisions in a single and consistent structure. It integrates investment strategies (also known as asset allocation strategies), liability decisions (e. g., borrowings) and savings strategies (or re-investment decisions) in a comprehensive fashion. As such, the system forms the basis for assessing and managing risks for large institutional organizations, including banks, savings and loans, insurance companies, pension plans, and government entities. Several noteworthy applications of stochastic programming for financial planning include the Russell–Yasuda investment system for insurance companies [6], the Towers Perrin investment system for pension plans [22], the integrated simulation and optimization system for the Metropolitan Life Insurance Company [35], and the integrated product management system [12]. In each case, asset investment decisions are combined with liability choices in order to maximize the investor's wealth over time.

We describe a generalized network model for multiperiod investment planning [23]. While some real-world issues are difficult to accommodate within the network context and must be handled as general linear constraints, the network provides a visual reference for the financial planning system. We divide the entire planning horizon T into two discrete time intervals T_1 and T_2 where $T_1 = \{0, \ldots, \tau\}$ and $T_2 = \{\tau + 1, \ldots, T\}$. The former corresponds to periods in which investment decisions are made. Period τ defines the date of the planning horizon; we focus on the investor's position at the beginning of period τ. Decisions occur at the beginning of each time stage. Much flexibility exists. An active trader might see his time interval as short as minutes, whereas a pension plan advisor will be more concerned with much longer planning periods such as the dates between the annual Board of Director's meeting. It is possible for the steps to vary over time — short intervals at the beginning of the planning period and longer intervals towards the end. T_2 handles the horizon at time τ by calculating economic and other factors beyond period τ up to period T. The investor cannot render any active decisions after the end of period τ.

Asset investment categories are defined by set $A = \{1, \ldots, I\}$, with category 1 representing cash. The remaining categories can include broad investment groupings such as stocks, bonds, and real estate. The categories should track well-defined market segment. Ideally, the co-movements between pairs of asset returns would be relatively low so that diversification can be done across the asset categories.

In our approach, uncertainty is represented by a number of distinct realizations. Each complete realization of all uncertain parameters gives rise to a *scenario*; we denote by S the discrete set of all scenarios. Several scenarios may reveal identical values for the uncertain quantities up to a certain period – i. e., they share common information history up to that period (see Fig. 1). Scenarios that share common information up to a specific period must yield the same decisions up to that period. We will address the representation of the information structure through a condition known as *nonanticipativity*.



Financial Optimization, Figure 1
Scenario tree

We assume that the portfolio is rebalanced at the beginning of each period. Alternatively, we could simply make no transaction except reinvest any dividend and interest – a buy and hold strategy. For convenience, we also assume that the cashflows are reinvested in the generating asset category and all the borrowing is done on a single period basis.

For each $i \in A$, $t \in T_1$, and $s \in S$, we define the following parameters and decision variables.

Parameters

- $r^s_{i,t} = 1 + \rho^s_{i,t}$, where $\rho^s_{i,t}$ is the percent return for asset i, time period t, under scenario s (projected by the stochastic modeling subsystem). π_s is the probability that scenario s occurs, $\sum_{s=1}^{S} \pi_s = 1$.
- w_0 is the wealth in the beginning of time period 0.
- $\sigma_{i,t}$ are the transaction costs incurred in rebalancing asset i at the beginning of time period t (symmetric transaction costs are assumed, i. e., cost of selling equals cost of buying).
- β^s_t is the borrowing rate in period t under scenario s.

Decision Variables

- $x^s_{i,t}$ is the amount of money in asset category i, in time period t, under scenario s, after rebalancing.
- $v^s_{i,t}$ is the amount of money in asset category i, in the beginning of time period t, under scenario s, before rebalancing.
- w^s_t is the wealth at the beginning of time period t, under scenario s.
- $p^s_{i,t}$ is the amount of asset purchased for rebalancing in period t, under scenario s.
- $d^s_{i,t}$ is the amount of asset i sold for rebalancing in period t, under scenario s.
- b^s_t is the amount borrowed in period t, under scenario s.

With these definitions in place, we can present the deterministic equivalent of the stochastic asset allocation problem.

Model SP

$$\max Z = \sum_{s=1}^{S} \pi_s f(w^s_\tau) \tag{1}$$

such that

$$\sum_i x^s_{i,0} = w_0, \quad \forall s \in S, \tag{2}$$

$$\sum_i x^s_{i,\tau} = w^s_\tau, \quad \forall s \in S, \tag{3}$$

$$\begin{aligned} &v^s_{i,t} = r^s_{i,t-1} x^s_{i,t-1}, \\ &\forall s \in S, \quad t = 1, \dots, \tau, \quad i \in A, \end{aligned} \tag{4}$$

$$\begin{aligned} &x^s_{i,t} = v^s_{i,t} + p^s_{i,t}(1 - \sigma_{i,t}) - d^s_{i,t}, \\ &\forall s \in S, \quad i \neq 1, \quad t = 1, \dots, \tau \end{aligned} \tag{5}$$

$$\begin{aligned} &x^s_{1,t} = v^s_{1,t} + \sum_{i \neq 1} d^s_{i,t}(1 - \sigma_{i,t}) \\ &- \sum_{i \neq 1} p^s_{i,t} - b^s_{t-1}(1 + \beta^s_{t-1}) + b^s_t \\ &\forall s \in S, \quad t = 1, \dots, \tau, \end{aligned} \tag{6}$$

$$\begin{aligned} &x^s_{i,t} = x^{s'}_{i,t} \\ &\text{for all scenarios } s, s' \\ &\text{with identical past up to time } t. \end{aligned} \tag{7}$$

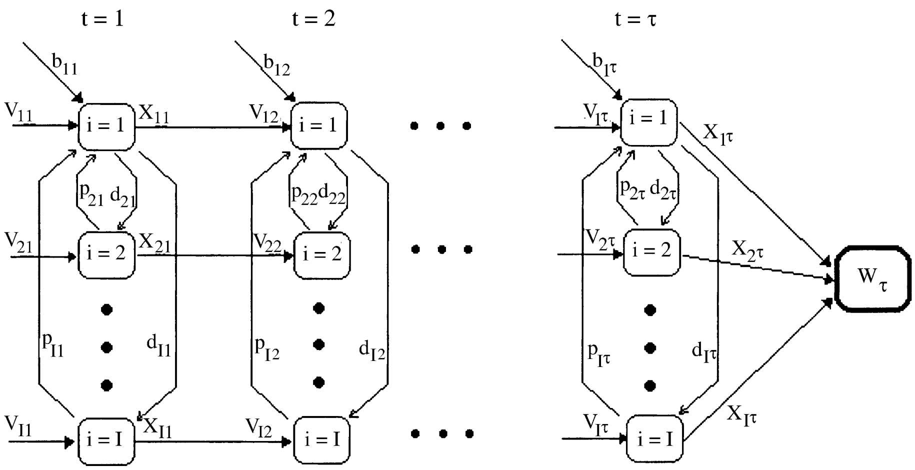
The generalized network model is presented in Fig. 2. The nonlinear objective function (1) can take several different forms. If the classical mean-variance function is employed, then (1) becomes $\max Z = \eta \operatorname{Mean}(w_\tau) - (1 - \eta) \operatorname{Var}(w_\tau)$, where $\operatorname{Mean}(w_\tau)$ is the average total wealth and $\operatorname{Var}(w_\tau)$ is the variance of the total wealth across the scenarios at the end of period τ. Parameter η indicates the relative importance of variance as compared with the expected value. This objective leads to an efficient frontier of wealth at period τ. An alternative to mean-variance is the von Neumann–Morgenstern expected utility of wealth at period τ. Here, the objective becomes

$$\max Z = \sum_{s=1}^{S} \pi_s \operatorname{Utility}(w^s_\tau),$$

where $\operatorname{Utility}(W)$ is the von Neumann–Morgenstern utility function [15]. The two objective functions are equivalent under certain conditions on the distribution of returns and the shape of the utility function [17].

Constraint (2) guarantees that the total initial investment equals the initial wealth. Constraint (3) represents the total wealth in the beginning of period τ. This constraint can be modified to include assets, liabilities, and investment goals. The modified result is called the *surplus wealth* [21]. Most investors make allocation decisions without reference to liabilities or investment goals. J.M. Mulvey employs the notion of surplus wealth to the mean-variance and the expected utility models to address liabilities in the context of asset allocation strategies. Constraint (4) depicts the wealth $v^s_{i,t}$ accumulated at the beginning of period t before rebalancing in asset i. The flow balance constraint for all assets except cash for all periods is given by constraint (5). This constraint guarantees that the amount invested in period t equals the net wealth for asset. Constraint (6) represents flow balancing constraint for cash. Nonanticipativity constraints are represented by (7). These constraints ensure that the scenarios with the same past will have identical decisions up to that period. While these constraints are numerous, solution algorithms take advantage of their simple structure.

Model SP is a split variable formulation of the stochastic asset allocation problem. This formulation has proven successful for solving the model using techniques such as progressive hedging algorithm of [27] and quadratic diagonal approximation of [25]. Split variable formulation is also found beneficial by direct solvers that use the interior point method [19]. By substituting constraint (7) back in constraint (2) to (6), we



Financial Optimization, Figure 2
Generalized network model for each scenario *s* ∈ *S*

obtain a compact formulation of the stochastic allocation problem. Constraints for this formulation exhibit a dual block diagonal structure. This formulation may be better for some direct solvers [19].

Scenario Generation

Scenario analysis offers an effective, and easily understood tool for addressing the stochastic elements in a multistage financial model. We define a scenario as a single deterministic realization of all uncertainties over the planning horizon. Ideally, the process constructs scenarios that represent the universe of possible outcomes. This objective differs from generation of a single scenario, say for forecasting and trading strategies. We are interested in constructing a 'representative' set of scenarios that are both optimistic and pessimistic within a risk analysis framework. Such an effort was undertaken by Towers Perrin (one of the largest actuarial firms in the world) using a system called CAP:Link [22]. The system entails a cascading of a set of submodels, starting with the interest rate component. Towers Perrin employs a version of the Brennan–Schwartz [5] two-factor interest rate model. The other submodels are driven by the interest rates and other economic factors. Towers Perrin has implemented the system in over 14 countries in Europe, Asia, and North America.

Scenario generation requires the estimation of the input parameters for the modeling of the economic factors. The ability to choose the 'correct' or 'best' set of parameters is essential if such models are to have practical value. A variety of techniques are available for estimating the economic factors required for projected returns and liabilities. See [24] and [1] for a discussion of some these techniques. See also [8,9] for a treatment of the robustness of scenario generation.

Solution Techniques

In this section we review a number of algorithms available to solve the asset allocation models. We focus on solutions to multistage stochastic programs possessing discrete-time decisions with a modest number of scenarios – typically under 1000 to 3000 – and nonlinear objective functions addressing risk aversion. The model's size depends on the number of decision variables and the form of the nonanticipativity rules. If Model SP is selected, the model becomes a convex program whose size hinges on the number of scenarios that are placed in *S*.

Direct Solvers

The simplest approach when the objective is linear is to use an efficient linear programming solver. Although simpler to handle, LP does not represent risk aversion well. See [19] for a solution of the multistage asset allocation problem with a linear objective function using OB1 and MINOS. OB1 is a primal-dual interior point algorithm for solving linear programs [18]. MINOS is a nonlinear programming code that can also solve LP [28]. On a compact formulation, MINOS outperformed OB1 on several test problems. The split formulation, however, significantly reduced the time required by OB1 to yield the fastest solution times.

When the objective is nonlinear, a general purpose nonlinear programming code can be used for solution. However, the nonlinear interior point methods have advantages over these codes. For example, in mean-variance applications, the covariance matrix can be factored to convert the mean-variance function into a separable function. This is achieved by a modest increase in the number of constraints. R.J. Vanderbei and T.J. Carpenter [34] show that nonlinear interior point methods can take advantage of the separable structure despite the increase in the number of constraints. A similar transformation is possible with the expected utility objectives as discussed in [3].

Primal-dual interior point algorithms can be specialized to solve nonlinear stochastic optimization problems. See [7] for an extension of a primal-dual interior point procedure for linear programs to the case of convex separable quadratic objectives. The extension is tested on the asset allocation problems of [26] and compared to MINOS. The primal-dual interior point method compared favorably with MINOS, especially for the larger test problems. In the direct solution of nonlinear programs via interior point methods, the primary computational step is the factorization of the normal equations ADA^{T}, where A is the coefficient matrix and D is a diagonal matrix [18]. This factorization is typically done by means of the Cholesky (LL^{T}) method. A major difficulty when applying these algorithms to stochastic optimization problems has to do with the sparsity structure of A. Several efficient approaches are now (2000) available to address the sparsity issue, the most recent being the tree dissection method [2].

Ideas of using parallel computing for stochastic programs have been around for quite some time [29]; [14]. More recently (1993), E. Rothberg [31] developed an extremely efficient method for carrying out sparse matrix factorization in a parallel environment. Rothberg's factorization coupled with tree dissection concepts provides some very encouraging results for stochastic programs. Initial evidence indicates that parallel direct solvers will be able handle stochastic programs with over 10,000 scenarios within several minutes of runtime in a parallel environment.

Decomposition Algorithms

Considerable progress has been made in the design of efficient decomposition algorithms for solving multistage stochastic programs. A number of decomposition algorithms are based on the augmented Lagrangian function, such as the progressive hedging algorithm (PHA) and the diagonal quadratic approximation (DQA). PHA applies to the variable split form of the multistage stochastic program. The nonanticipativity constraints are placed in the objective function as penalty and multiplier terms, and are progressively enforced by an iterative procedure. Mulvey and H. Vladimirou [27] compare the performance of the progressive hedging algorithm to alternative solution strategies on a set of linear and nonlinear portfolio management problems. The general purpose optimizer MINOS [28] solve these test problems in their compact form. This is the most efficient program formulation for MINOS because it results in the smallest constraint matrix, i. e. the size of the basis is minimized. The linear problems were also solved using the primal-dual interior code (OB1) of [18]. For nonlinear test cases, they employ an extension of the primal-dual interior point method to convex, separable optimization programs [7]. The staircase formulation obtained by partial variable splitting is employed in these terms. On linear problems the progressive hedging algorithm was faster than MINOS. It was also faster than OB1 when the compact form was used. Interior point outperformed PHA for staircase structures. On nonlinear problems, PHA maintains its superiority over MINOS, particularly on large test problems. The progressive hedging algorithm also fares well against interior point algorithm on nonlinear problems, outperforming it in several cases.

DQA forms an augmented Lagrangian function by dualizing nonanticipativity constraints. The DQA algorithm approximates the Lagrangian at the current iterate by a quadratic and separable term [25]. The outer loop revises the dual variable by the method of multipliers, whereas the inner loop consists of separable quadratic or convex terms. DQA is a flexible scheme which can be implemented in many ways, in particular, in a parallel distributed environment. Mulvey and A. Ruszczynski [25] compare the performance of DQA with highly specialized methods for linear two-stage problems. The most successful methods found so far are based on Benders decomposition, suggested for stochastic programming in [33]. MSLiP [11] is a recent (1990) implementation of this idea, which allows for solving linear multistage problems in a nested formulation. Mulvey and Ruszczynski [25] show that the specialized decomposition techniques MSLiP and DQA outperform MINOS.

Conclusions and Future Directions

The proposed multistage financial planning model provides a general framework for integrating all asset and liability decisions for a large financial entity – such as an insurance company, bank, or pension plan – as well as for individual investors. This comprehensive approach measures the risk and rewards of alternative investment strategies. Without an integrative asset-liability model, investors are unable to properly measure risks to their wealth. The usual asset-only approach inadequately evaluates the impact of investments on wealth and achieving investment goals. The main lesson is that investment models must be tailored to individual circumstances. The multistage stochastic program provides an ideal vehicle for developing a financial plan that fits the investor's needs.

Future research should continue along several dimensions. First, we must increase the size of solvable stochastic programs so that additional scenarios can be handled in a practical fashion. There is no fundamental reason why we cannot address 10,000 to 100,000 scenarios using parallel and distributed computers. Certainly, the raw computing power will be available. Whether or not direct solvers or decomposition algorithms are best is a matter for future research.There are a number of algorithmic items to explore. One is to take further advantage of the structure of the multistage stochastic program within a parallel interior-point algorithm. For instance, we can conduct the Cholesky factorization using modern sparse matrix calculations on parallel or distributed computers. Rothberg's approach [31] seems to be a potential winner. Solving the stochastic program as quickly as possible will increase the chances that individual investors and institutions will apply the models. In the case of decomposition methods, the sparse matrix calculations are key for techniques such as DQA which use an interior-point algorithm for solving subproblems. Any substantial progress on this issue leads to immediate gains in the decomposition algorithm. Also, the restarting issue for interior-point algorithms remains.

Another computational issue involves generating scenarios. In particular, out-of-sample testing will be critical in order to compute valid bounds on the model recommendations. When it applies, dynamic stochastic control can be useful. The control system assists in the selection of the scenarios – for instance, by generating importance estimates for adding (or deleting) scenarios as they affect the solution to the control problem. These scenarios should be linked to the stochastic program. Of course, embedding a stochastic program within a simulation system such as carried out in [35] to evaluate the precision of the recommendations is possible. The approach requires large computational resources and may be impractical. Linking simulation and optimization models, however, will be increasingly important, as multistage stochastic programs become more widespread in practice.

A third issue deals with the automatic calibration of scenario generation systems using a nonlinear program. For example, the two-factor interest rate model possesses seven parameters, including the correlation coefficient for the Weiner terms. Setting these parameters requires considerable effort. There are several competing objectives: minimizing deviations on the summary statistics with respect to historical values; meeting expectations regarding future asset returns such as stocks and bonds; and avoiding trends that are clearly unrealistic. The estimation approaches developed in [24] and [1] address some these issues, but more work is needed to fully understand both modeling and computational issues of automatic calibration of scenarios.

See also

- ▶ Competitive Ratio for Portfolio Management
- ▶ Financial Applications of Multicriteria Analysis
- ▶ Portfolio Selection and Multicriteria Analysis
- ▶ Robust Optimization
- ▶ Semi-infinite Programming and Applications in Finance

References

1. Berger AJ, Mitchell JE, Mulvey JM, Rush R (1997) A tabu search procedure for target-matching in financial scenario generation. Techn. Report Dept. Civil Engin. Oper. Res. Princeton Univ. SOR-97-16
2. Berger AJ, Mulvey JM, Rothberg E, Vanderbei R (1996) Solving multi-stage stochastic programs using tree dissection. Techn. Report Dept. Civil Engin. Oper. Res. Princeton Univ. SOR-95-07
3. Berger AJ, Mulvey JM, Ruszczyński A (1994) An extension of the DQA algorithm to convex stochastic programs. SIAM J Optim 4:735–753
4. Birge JR (1997) Stochastic programming computation and applications. INFORMS J Comput 9:111–133
5. Brennan MJ, Schwartz ES (1982) An equilibrium model of bond pricing and a test of market efficiency. J Financial and Quantitative Anal 17:75–100
6. Cariño DR, Kent T, Myers DH, Stacy C, Sylvanus M, Turner A, Watanabe K, Ziemba WT (1994) The Russell–Yasuda Kasai model: An asset liability model for a Japanese insurance company using multi-stage stochastic programming. Interfaces 24:29–49
7. Carpenter TJ, Lustig IJ, Mulvey JM, Shanno DF (1993) Separable quadratic programming via a primal-dual interior point method and its use in a sequential procedure. ORSA J Comput 5:182–191
8. Dupacova J (1987) Stochastic programming with incomplete information: A survey of results on postoptimization and sensitivity analysis. Optim 18:507–532
9. Dupacova J (1996) Scenario based stochastic programs: Resistance with respect to sample. Ann Oper Res 64:21–38
10. Elton EJ, Gruber MJ (1995) Modern portfolio theory and investment analysis, 5th edn. Wiley, New York
11. Gassmann HI (1990) A computer code for the multi-stage stochastic linear programming problem. Math Program 47:407–423
12. Golub B, Holmer M, McKendall R, Pohlman L, Zenios SA (1995) A stochastic programming model for money management. Europ J Oper Res 85:282–296
13. Jarrow R, Maksimovic V, Ziemba W (1995) Asset and liability allocation in a global environment. In: Mulvey JM, Ziemba WT (eds) Handbook Oper. Res. and Management Sci.: Finance. Elsevier, Amsterdam
14. Jessup ER, Yang D, Zenios SA (1994) Parallel factorization of structured matrices arising in stochastic programming. SIAM J Optim 4:833–846
15. Keeney R, Raiffa H (1993) Decisions with multiple objectives: Preferences and value Tradeoffs. Cambridge Univ. Press, Cambridge. First printed by Wiley, 1976
16. Konno H, Luenberger DG, Mulvey JM (eds) (1993) Ann. Oper. Res. Baltzer, Basel
17. Kroll Y, Levy H, Markowitz H (1984) Mean variance versus direct utility Maximization. J Finance 39:47–62
18. Lustig IJ, Marsten RE, Shanno DF (1994) Interior point methods for linear programming: Computational state of the art. ORSA J Comput 6:1–14
19. Lustig IJ, Mulvey JM, Carpenter TJ (1991) Formulating stochastic programs for interior point methods. Oper Res 39:757–770
20. Markowitz HM (1991) Portfolio selection: Efficient diversification of investments, 2nd edn. Blackwell, Oxford
21. Mulvey JM (1989) A surplus optimization perspective. Investment Management Rev 3:31–39
22. Mulvey JM (1995) Generating scenarios for the Towers Perrin investment systems. Interfaces 26:1–15
23. Mulvey JM, Rosenbaum DP, Shetty B (1997) Strategic financial risk management and operations research. Europ J Oper Res 97:1–16
24. Mulvey JM, Rosenbaum DP, Shetty B (1999) Stochastic scenario generation systems. Europ J Oper Res 118:563–577
25. Mulvey JM, Ruszczyński A (1995) A new scenario decomposition method for large-scale stochastic optimization. Oper Res 43:477–490
26. Mulvey JM, Vladimirou H (1989) Stochastic network optimization models for investment planning. Ann Oper Res 20:187–217
27. Mulvey JM, Vladimirou H (1992) Stochastic network programming for financial planning problems. Managem Sci 38:1642–1664
28. Murtagh BA, Saunders MA (1987) MINOS 5.1 user's guide. SOL Techn. Report Systems Optim. Lib. Stanford Univ. 83-20R
29. Nielsen S, Zenios SA (1993) A massively parallel algorithm for nonlinear stochastic network problems. Oper Res 41:319–337
30. Perold AF (1984) Large-scale portfolio optimization. Managem Sci 30:1143–1160
31. Rothberg E (1993) Performance of panel and block approaches to sparse Cholesky factorization on the iPSC/860 and Paragon multicomputers. Techn. Report Intel Supercomputer Systems Div., Beaverton, OR
32. Speranza MG (1993) Linear programming models for portfolio optimization. J Finance 14:107–123
33. Van Slyke R, Wets RJ-B (1969) L-shaped linear programs with applications to optimal control and stochastic programming. SIAM J Appl Math 17:638–663
34. Vanderbei RJ, Carpenter TJ (1993) Symmetric indefinite systems for interior-point methods. Math Program 58:1–32
35. Worzel K, Vassiadou-Zeniou C, Zenios SA (1995) Integrated simulation and optimization models for tracking fixed-income indices. Oper Res 42:223–233
36. Zenios SA (ed) (1993) Financial optimization. Cambridge Univ. Press, Cambridge
37. Zenios SA, Ziemba WT (1992) Theme issue on financial modeling. Managem Sci 38:1525–1528
38. Ziemba WT, Mulvey JM (eds) (1998) Worldwide asset and liability modeling. Cambridge Univ. Press, Cambridge

Finite Complete Systems of Many-valued Logic Algebras

LADISLAV J. KOHOUT
Department Computer Sci., Florida State University,
Tallahassee, USA

MSC2000: 03B50, 68T15, 68T30

Article Outline

Keywords
Boolean 2-Valued Logic Algebras
A Repertory of Complete Many-Valued Logic Normal Forms
Many-Valued Families of the Pinkava Logic Algebras
General Pi-Algebras
Families of Pi-Algebras and their Functionally Complete Normal Forms
The Taxonomy of the PI-Algebras of Many-Valued Logics
Special Subfamilies of n-Valued PI-Systems
PI-Algebras and a New Variety of 2-Valued Normal Forms
Theoretical and Practical Importance of PI-algebras
The Requirement of Functional Completeness
Satisfiability Problem in Computational and Descriptive Complexity Theory
Central Importance of the Satisfiability Problem of Boolean Formulas in Complexity Theory
Open Problems in the Complexity Theory of PI-algebras (1999)
Applications of PI-Algebras
See also
References

Keywords

Many-valued logics; PI-logic algebras; Discrete functions; Normal form; Functional completeness; Satisfiability; Computational complexity; Descriptive complexity; Scientific applications; Medical applications

First we review briefly some facts about 2-valued discrete functions (two-atom Boolean algebras). Then we proceed with various n-valued extensions and generalizations which are not necessarily always Boolean. The most general class of systems to be discussed are Pi-algebras. The logic connectives of these algebras are partial nonassociative noncommutative general algebraic *groupoids* [13,30].

Associative connectives, such as various families of t-norms and t-conorms [10] which are widely used in fuzzy logics are the special instances of PI-algebra groupoid connectives. References to some applications of Pi-logic algebras conclude this entry.

Although the primitives of PI-algebras are only partially defined they are *functionally complete*. Hence one can represent any finite discrete function by PI-normal forms.

Definition 1 A finite *discrete function* of k arguments $f(x_1, \dots, x_k)$ is a mapping from the k-fold Cartesian product of a set A to itself. In symbols: $f\colon A \times \cdots \times A \to A$, where A is a finite set containing n elements, $A = \{a_0, \dots, a_{n-1}\}$.

For typographic convenience, we shall map the elements of A into the finite subset $\mathbf{N}$ of nonnegative numbers by the assignment $a_i = i$, namely $A = \{0, \dots, n-1\}$. This does not imply that the ordering of natural numbers is always relevant to our algebraic considerations. These numbers should just be considered as more convenient labels than, say, a_i for the elements of a finite set A.

Boolean 2-Valued Logic Algebras

A *Boolean 2-valued function* is a discrete function that takes its values from the two-element set $\{0, 1\}$. We can form 16 different two-argument functions on the set $\{0, 1\}$. The ten nontrivial of these are shown below.

Finite Complete Systems of Many-valued Logic Algebras, Table 1
Two-argument connectives of the 2-valued logic

x	y	$\wedge$	$\vee$	$\rightarrow$	$\leftarrow$	$\equiv$	$\oplus$	$\nrightarrow$	$\nleftarrow$	$\downarrow$	$\mid$
0	0	0	0	1	1	1	0	0	0	1	1
0	1	0	1	1	0	0	1	1	0	0	1
1	0	0	1	0	1	0	1	0	1	0	1
1	1	1	1	1	1	1	0	0	0	0	0

Some of these operations, also called *logic algebra connectives*, play an important role in logic. Hence they are given special names to stress their meaning and significance.

When '1' is interpreted as 'True' and '0' as 'False', then ∨ represents logical AND, ∧ (nonexclusive) logical OR, ⊕ exclusive logical OR (i. e. XOR). The connective ≡ is *logical equivalence* which captures the equivalence of two propositions, and → is an *implication operator* which captures the validity (truth-value) of the *conditional* 'If __ then __'.

The connectives, together with some inference rules (e. g. modus ponens, see ► Checklist paradigm semantics for fuzzy logics), make a system of classical propositional logic.

Let us recall that any Boolean function can be expressed by the conjunctive normal form (CNF) or disjunctive normal form (DNF). Let us introduce the notation $x^\sigma = (x \wedge \sigma) \vee (\overline{x} \wedge \overline{\sigma})$, where $\overline{x}$ denotes the *negation* of x, and σ is a parameter equal either to 0 or 1. Then it is obvious that

$$x^\sigma = \begin{cases} \overline{x} & \text{when } \sigma = 1, \\ x & \text{when } \sigma = 0. \end{cases}$$

x^σ is called *literal*.

Theorem 2 (Normal forms theorem) *Every Boolean function $f(x_1, \ldots, x_n)$ can be represented by their canonical (full) CNF and DNF normal forms.*

i) The disjunctive normal form:

$$\bigvee_{f(\sigma_1,\ldots,\sigma_n)=1} x_1^{\sigma_1} \wedge \cdots \wedge x_n^{\sigma_n}.$$

ii) The conjunctive normal form:

$$\bigwedge_{f(\sigma_1,\ldots,\sigma_n)=0} x_1^{\overline{\sigma_1}} \vee \cdots \vee x_n^{\overline{\sigma_n}}.$$

A *clause* in a DNF consists either of a literal or of a conjunction of literals. In a CNF, on the other hand, a clause consists either of a literal or of a disjunction of literals.

Because we can express any Boolean function by formulas formed by means of the sets of connectives CNF-Cset (= {∧, ∨, ¬}), DNF-Cset (= {∨, ∧, ¬}), we call these sets *complete sets* of connectives.

A Repertory of Complete Many-Valued Logic Normal Forms

Important structural relationships that provide the algebraic backbone of various logics are contained in their normal forms. It is possible to generalize from two-valued normal forms to *many-valued normal forms* in various ways. We shall discuss here one such generalization, namely partial functionally complete Pinkava algebras (Pi-algebras) which offer some interesting insights and have also a significant practical value. The systems were discovered in 1971 by V. Pinkava [24,25] as a significant generalization of the systems used by Pinkava in his previous work [21].

Many-Valued Families of the Pinkava Logic Algebras

Definition 3 ([30,35]) The *Pinkava n-valued family of logical calculi* Pi = {A, □, ⋄, ⊙, ↪} consists of the partially defined connectives operating on the value-set {0, …, n −1}:

$$v_i \Diamond v_j = \begin{cases} 0 & \text{if } v_i = 0, \\ 1 & \text{if } v_i \neq 0 \;\&\; v_j = 1, \\ \text{undefined} & \text{otherwise}, \end{cases}$$

$$v_i \Box v_j = \begin{cases} 0 & \text{if } v_i = 0, \\ v_j & \text{if } v_i \neq 0 \;\&\; v_j = 1, \\ \text{undefined} & \text{otherwise}, \end{cases}$$

$$v_1 \odot v_2 = \begin{cases} v_j & \text{if } v_i = 0, \\ \text{undefined} & \text{otherwise}, \end{cases}$$

$$v^{\hookrightarrow} = v + 1 \pmod{n},$$

$$\Psi_\kappa = \begin{cases} 1 & \text{if } v = \kappa, \\ 0 & \text{if } v \neq \kappa, \end{cases}$$

where $i, j \in \{1, 2\}$.

Note that the carrier and the *characteristic functions* can also be generated in the Pinkava logic calculi by the connectives. For example, the characteristic function Ψ can be generated by ⋄, [13].

Theorem 4 (Complete normal forms) *Any n-valued logic function that is obtained by a completion of the partially defined Pinkava connectives of the type {⊙ □, ⋄, ↪} defined above is functionally complete and can be* *expressed in the following canonical normal form:*

$$f(v_1,\dots,v_n) = \bigodot_{f(v_1,\dots,v_n)\neq 0} \left[c_\gamma \Box \left(\Diamond_{s=1}^{n} \psi_{\alpha_s}(v_s)\right)\right].$$

General Pi-Algebras

Families of Pi-Algebras and their Functionally Complete Normal Forms

The Pinkava logic calculi can be further generalized [11,13,14]. These generalizations are called *Pi-algebras*. The connectives involved are partial nonassociative noncommutative general (algebraic) groupoids in their most general form. *Associative connectives*, such as the *t-norms* and *t-conorms* [10,44] are special instances of them (see also ► Boolean and fuzzy relations).

Definition 5 (*Families of Pi-algebras*) Let Pi be an algebra with carrier P such that PI = $\langle P, \diamond, \Box, \odot, \Phi\rangle$, where [13]:

1) $\langle P, \diamond\rangle$ is an arbitrary groupoid with zero $z_\diamond$, without divisors of zero, and with the almost absorbing element $a_\diamond$ such that $a_\diamond \diamond p = p \diamond a_\diamond = a_\diamond$ for every $p \in P, p \neq z_\diamond$.
2) $\langle P, \odot\rangle$ is an arbitrary groupoid with unit $e_\odot$.
3) $\langle P, \Box\rangle$ is an arbitrary groupoid with a right zero $z_{r\,\Box}$ and a right unit $e_{r\,\Box}$.
4) Φ is a discrete *cyclic shift function* Φ:$P \to P$ satisfying the following conditions: Given a discrete cyclic order of P, then for every $p \in P$ it holds that $p \preccurlyeq \Phi(p)$ and $\Phi^0(p) = p$, $\Phi^{k+1}(p) = \Phi(\Phi^k(p))$.

In the above definition $a \preccurlyeq b$ means that a is a direct predecessor of b.

Definition 6 Let $p_1, p_2 \in P$, and Φ be a cyclic shift function. Then the *advance* δ from p_1 to p_2 with respect to Φ is the least ordinal such that $\Phi^\delta(p_1) = p_2$. We write $\delta_\Phi(p_1) = p_2$. The advance δ^* denotes the inverse of δ.

Theorem 7 (Canonical normal forms) *Let the advances δ_1, δ_2 be defined by the formulas $\delta_1 := (a_\diamond, e_{r\,\Box})$, $\delta_2 := (z_\Box, e_\odot)$. Then any function f on the carrier P in a Pi-algebra can be expressed in its canonical normal form:*

$$\begin{aligned} &f(v_1, v_2, \dots) \\ &\quad= \bigodot_{f\neq e_\odot} \Phi^{\delta_2} \left\{\Phi^{\delta_2^*}(c_\gamma)\Box\left(\Diamond_{s=1}^{\operatorname{card} P} \Phi^{\delta_1}[\Psi_{\alpha_s}(v_s)]\right)\right\}. \end{aligned}$$

The argument scope of the outer connective of the normal form is $\odot_{\{\forall f(v_1,v_2,\dots):f\neq(e_\odot)\}}$. This means that the values $e_\odot$ are omitted.

Theorem 8 (Functional completeness) *Any Pi-algebra is functionally complete if, given the advance $\delta_1 = \delta(a_\diamond, e_{r\,\Box})$, it also holds that $\delta_1 = \delta(z_\diamond, z_\Box)$.*

Theorem 9 *If the right zero $z_{r\,\Box}$ is also the zero and the right unit $e_{r\,\Box}$ is also the unit of the groupoid $\langle P, \Box\rangle$, then the following normal form is also functionally complete:*

$$\begin{aligned} &f(v_1, v_2, \dots) \\ &\quad= \bigodot_{f\neq e_\odot} \Phi^{\delta_2} \left\{\Phi^{\delta_2^*}(c_\gamma)\Box\left(\Box_{s=1}^{n} \Phi^{\delta_1}[\Psi_{\alpha_s}(v_s)]\right)\right\}. \end{aligned}$$

The Taxonomy of the PI-Algebras of Many-Valued Logics

The main theoretical question that the PI-algebras answer is: 'Which features of two-valued Boolean logic structures disappear and which are preserved and carried over into the extensions and generalizations to many-valued logics'? The Pi-logic algebras are partial systems that put under one roof a wide variety of families of functionally complete many-valued logical systems. Thus they offer a useful framework in which various generalizations and extensions can be carried out. They also provide a sound base for a useful *classification of many-valued logics* by their various properties. This approach, based on *PI-normal forms*, usefully complements another way of classifying the many-valued logic connectives by groups of logic transformations (see ► Checklist paradigm semantics for fuzzy logics).

Special Subfamilies of *n*-Valued PI-Systems

Because the Pinkava connectives are only partially defined, it is possible by imposing further restrictions on these connectives to define subclasses of the functionally complete Pinkava systems. For example, consider the restrictions

a) $(v_1^{\hookrightarrow} \Box v_2^{\hookrightarrow})^{\hookrightarrow n-1} = v_1 \odot v_2$;
b) $(v_1^{\hookrightarrow n-1} \odot v_2^{\hookrightarrow n-1})^{\hookrightarrow} = v_1 \Box v_2$.

They make the subclass $\{\diamond, \Box, \hookrightarrow\}$ functionally complete. Imposing some other restrictions we can obtain other subclasses of functionally complete connectives. For example, the *Post system*, the *Aizenberg–Rabinovich system*, the *Zhegalkin algebra*, and *lattice-type many-valued logic systems* can be so obtained. For further details see e. g. [13,38]. A partial taxonomy of various subclasses of the Pinkava systems can be found in [13, Fig. 8.1] or in [11 p. 279, Fig. 1]. Several important subclasses of the Pinkava systems are presented in [26]. For the subclasses that are generalizations of the *Sheffer function*, see [32]. The systems particularly suitable for *minimizations* appear in [33].

PI-Algebras and a New Variety of 2-Valued Normal Forms

It is illuminating to look at two-valued well-known special instances of logic connectives and classify them in terms of Pi-algebra connective types. This reveals that there are other canonical normal forms in addition to DNF and CNF. For instance, the $\odot$, which is partial, offers two distinct completions: either Boolean (inclusive) OR $\vee$ or exclusive-OR (nonequivalence) $\oplus$. Although the connectives $\Box$ and $\diamond$ are identical in the two-valued case, both forming Boolean $\wedge$, they extend each to a distinct partial connective for $n > 2$. This is because each of these connectives plays a different role in the normal form, serving a different purpose.

In order to explore more fully the richness of Pi-algebras, one has to look at their *taxonomy* in the general many-valued case. For a more detailed taxonomy see [13, Fig. 8.1.2] or [11].

Two highlights emerge from this approach:

1) Even in the simple two-valued case, the normal forms of generalized Pi-algebras subsume not only the conjunctive and disjunctive normal form but also the implication, equivalence, exclusive-or and other normal forms in one unifying pattern.
2) Two distinct general n-valued connectives may 'collapse' into a single connective when one sets $n = 2$. Viewed the other way, a two-valued connective may 'bifurcate' into two distinct types of connectives when more than two values are used. This bifurcation of structures and concepts is an interesting phenomenon that accompanies *fuzzification* of two-valued structures.

Theoretical and Practical Importance of PI-algebras

The Requirement of Functional Completeness

The functional completeness is of primary interest to a scientist or an engineer engaged in practical applications of many-valued logic. In such applications it is often desirable to have the means for generating all possible finite discrete functions by means of a complete set of many valued logic connectives.

For example, it is desirable to have a set of *logic gates* that can generate any *combinatorial switching circuit*. In pattern recognizers implemented by many-valued logic networks, the set of basic *'cognitive elements'* has to be complete, otherwise some patterns may be misclassified. The completeness is necessary in order to have the means for representation of all the possible discrete functions over a finite set of elements.

Similarly, in biological or psychological and medical models based on abstract classification of patterns by logic nets the choice of an incomplete set of connectives as the representational base of the model might yield a bias towards assumptions that are not contained in the experimental data. For example, in ethological models of instincts the representation using an incomplete set of connectives would represent the a priori assumption that certain forms of instincts do not exist. Yet the data might contain the evidence for these, but this evidence is not representable and will be discarded by an unfortunate choice of the incomplete set of connectives. In models of neuro-psychological disorders this might cause a priori exclusion of some impairments of the substratum structures, diminishing the predictive usefulness of such models.

The complexity of the normal forms as well as the complexity of the minimized many-valued logic (MVL) expressions depends on the character of the discrete function (i. e. the data) to be represented, the choice of an appropriate many-valued logic system of connectives, and the number of the discrete values of the value set A. Hence, only by the choice of an appropriate MVL system may we achieve an optimal representation in each specific application domain.

The choice of a suitable system is usually an iterative process, which requires a comparative evaluation of several systems, performed in order to optimize the choice. In order to assess whether a chosen system is functionally complete, a set of conditions sufficient to determine the completeness is required. Alternatively, a set of rules has to be given that would make it possible to generate complete systems of required additional properties directly. The constructive conditions for completeness given above for the normal forms of PI-algebras are such rules.

Other conditions for completeness of the same of greater generality are not so suitable for this purpose because the number of conditions necessary to test for completeness increases rapidly with increasing number of values *n* of a many-valued logic system. E. Post was first to give the general conditions for completeness of 2-valued logics ($n = 2$). These were later generalized by S.W. Jablonskij (S.V. Yablonskii) [9], J. Slupecki [45], A. Salomaa [43] and others. The most general conditions known at present are those given by I. Rosenberg [41,42] which are the generalization of the *Post conditions* for any *n*-valued finite case.

In all these later cases (unlike for PI-algebras initiated by Pinkava), the number of conditions increases astronomically with increasing *n*. For $n = 2$ (Post) there are 5 conditions that the logic system has to satisfy. For $n = 3$ (Jablonskij) there are 18 conditions. For a seven-valued MVL system ($n = 7$), there are 7,848,984 conditions to be tested. The general formula for any finite $n \leq 2$ is given by Rosenberg in [42]. This formula shows that, for large *n*, the number is rather prohibitive, hence of no practical value. On the other hand, PI-algebras can generate an infinite number of finite functionally complete systems of connectives for any finite *n*. This is so because the Pinkava complete sets of connectives are only partially specified and the completion of the 'blanks' by any values does not invalidate their completeness.

Satisfiability Problem in Computational and Descriptive Complexity Theory

Central Importance of the Satisfiability Problem of Boolean Formulas in Complexity Theory

The main goal in the *complexity theory of algorithms* is to distinguish problems that can be solved efficiently from those that cannot be. A computational solution to a problem is *practically feasible* if it belongs to the *complexity class P*, that can be computed by a deterministic algorithm in time bounded by a polynomial function of the size of the input data.

The central problem of complexity theory in computer science and a major problem of contemporary logic and mathematics is whether the class *P* is equal to the class *NP*. Problems solvable by nondeterministic algorithms in polynomial time belong to the class *NP*.

Problems in the class *NP* are computationally tractable only if they are of *polynomial complexity*, that is if $P = NP$.

A successful proof of the conjecture that $P \neq NP$ would on the other hand indicate that the *NP* class is of computationally not tractable *exponential complexity*.

The class *NP* contains many practical problems that can be characterised by the following property: There is no known way to compute a solution in polynomial time, but there is a known way to check in polynomial time whether a potential (e. g. guessed) solution is an actual solution.

The *satisfiability problem* [19,20] that concerns Boolean formulas [5] is closely related to the question about computational complexity of many other computational problems [6,7].

We say that a Boolean formula is *satisfiable* if there exists at least one way of assigning values to its variables so as to make it true. Finding the answer 'yes' or 'no' to this question is called the [2]. If the Boolean formula of our concern is written solely in the CNF we have the *SAT-CNF problem*. *SAT-k-CNF* is obtained by restricting SAT-CNF to Boolean formulas in *k*-CNF, where *k-CNF* is composed of clauses, each of which contains at most *k* literals [2].

It follows from *Cook's theorem* [3] that the question whether or not $P = NP$ is equivalent to asking whether there is a *polynomial time deterministic algorithm* (PTDA) recognizing the set of satisfiable Boolean propositional formulas (the SAT problem), or equivalently, a PTDA recognizing the set of propositional tautologies *TAUT* [19]. This demonstrates the central importance of the SAT problem for computational complexity theory.

In 1971 S.A. Cook [3] proved that every problem $X \in NP$ is *polynomially Turing reducible* [1,2] to the question about the complexity of *TAUT-DNF*, i. e. the set of propositional tautologies coded in DNF. In symbols: $X \leq_T^P$ TAUT-DNF.

This is related to one of the open (as of 1999) fundamental problems of logic [19]: Is there a *propositional proof system P* in which every tautology has a polynomial size proof? At present it is known only [4] that there exists a propositional proof system in which every tautology has a polynomial size of proof if and only if $NP = \text{co } NP$, i. e. the class NP is closed under complementation.

The relation $\leq^p_T$ is a pre-order (see ► Boolean and fuzzy relations) hence it provides the means for comparing the relative computational difficulty of problems [1]. Because it is a pre-order, it may contain various equivalence classes (see ► Boolean and fuzzy relations) of problems.

The statement: 'The SAT problem is NP-complete' is referred to as the *Cook–Levin theorem* in the literature. Using the reducibility relation $\leq^p_T$ together with this theorem yields a useful technique for providing proofs of the NP-completeness of other problems. We say that a problem X is *NP-complete* [1,2] if

- $X \in NP$; and
- $Y \leq^p_T X$ for every problem $Y \in NP$.

There is a great number of computational problems in the graph and set theories, the NP-completeness of which can be proven by reducing the SAT problem directly or indirectly to each of them. For example *dominating set*, *vertex cover*, *clique*, *3-SAT*, *3-colorability* [20]. SAT can be reduced to the clique problem, which in turn is reducible to the vertex cover problem. The vertex cover problem is reducible to the dominating set problem. Similarly, there is another chain of reductions: SAT to 3-SAT to 3-colorability. These reduction chains form a part of the semilattice generated by the reducibility relation.

Open Problems in the Complexity Theory of PI-algebras (1999)

Computational complexity of PI-logic algebras and normal forms is an uncharted territory. There is an infinite number of ways in which the partially specified but functionally complete PI-logic normal forms can be made fully specified, and a large variety of algebraic restrictions that can be placed upon them to generate particular fully defined systems.

Despite of their partial nature, the PI-normal form have well defined length even before their algebraic properties are completely specified. Hence one may expect that they will play some role in placing the upper bound on descriptive complexity [8] of propositional systems. This may be a promising direction of research in the future. It should also be noted that the generalized PI-normal forms allow for description of transformations from lattice based connectives to ring based connectives. Indeed, both are special instances of PI-logic algebras (see Theorem 7 and [13]). That might help to build a bridge between methods for analysis of algebraic propositional systems [40] with notions of descriptive complexity [8].

Applications of PI-Algebras

In addition to their theoretical significance, the functionally complete PI-systems have found a number of practical applications in various fields: in medicine, clinical behavioral sciences and neurology [12,22,23,34,36]; in data analysis and classification [37,39]; analysis of logical paradoxes [29,38]; automata theory and systems science [17,31,38]; design of MVL-switching circuits [11]; in dynamic computer protection also applicable to distributed an parallel systems [13,15,16]; logic and theorem proving [18,35]; optimization of discrete functions [27,33] and fuzzy logics [28].

See also

► Alternative Set Theory
► Boolean and Fuzzy Relations
► Checklist Paradigm Semantics for Fuzzy Logics
► Inference of Monotone Boolean Functions
► Optimization in Boolean Classification Problems
► Optimization in Classifying Text Documents

References

1. Balcázar JL, Díaz J, Gabarró J (1988) Structural complexity, vol I. Springer, Berlin
2. Brassard G, Bratley P (1996) Fundamentals of algorithmics. Prentice-Hall, Englewood Cliffs, NJ
3. Cook SA (1971) The complexity of theorem proving procedures. In: Proc. 3rd Annual ACM Symp. Theory of Computing. ACM, New York, pp 151–158

4. Cook SA, Reckhow AR (1979) The relative efficiency of propositional proof systems. J Symbolic Logic 44(1):36–50
5. Delgrande JP, Gupta A (1996) The complexity of minimum partial truth assignments and implication in negation-free formulae. Ann Math Artificial Intelligence 18(1):51–67
6. Du D-Z, Gu J, Pardalos PM (1997) Satisfiability problem: Theory and applications. Amer. Math. Soc., Providence, RI
7. Hochbaum DS (1997) Approximation algorithms for NP-hard problems. PWS, Boston, MA
8. Immerman N, Kolaitis PG (eds) (1997) Descriptive complexity and finite models. Amer. Math. Soc., Providence, RI
9. Jablonskij SV (1958) Funkcional'nye postrojenija v k-značnoj logike. Trudy Mat Inst Steklov 51:5–142
10. Klement EP, Mesiar R (1997) Triangular norms. In: Mesiar R, Riečan B (eds) Fuzzy Structures – Current Trends, Tatra Mountains Math. Publ. (Special Issue), vol 13. Math. Inst. Slovak Acad. Sci., Bratislava, pp 169–194
11. Kohout L (1974) The Pinkava many-valued complete logic systems and their application to the design of many-valued switching circuits. In: Rine DC (ed) Proc. 1974 Internat. Symp. Multiple-Valued Logic (West Virginia Univ., May, 1974). IEEE, New York, pp 261–284
12. Kohout LJ (1976) Application of multi-valued logics to the study of human movement disorders. In: Proc. Sixth Internat. Symp. Multiple-Valued Logic. IEEE, New York, pp 224–232
13. Kohout LJ (1990) A perspective on intelligent systems: A framework for analysis and design. Chapman and Hall and v. Nostrand, London–New York
14. Kohout LJ (1995) Epistemological aspects of many-valued logics and fuzzy structures. In: Höhle U, Klement EP (eds) Non-Classical Logics and their Applications to Fuzzy Subsets: A Handbook of Mathematical Foundations of Fuzzy Set Theory. Kluwer, Dordrecht, pp 291–339
15. Kohout LJ, Gaines BR (1975) The logic of protection. In: Lecture Notes Computer Sci, vol 34. Springer, Berlin, pp 736–751
16. Kohout LJ, Gaines BR (1976) Protection as a general systems problem. Internat J General Syst 3:1–21
17. Kohout LJ, Pinkava V (1976) The functional completness of Pi-algebras and its relevance to technological applications and biological modelling. In: Mamdani EH, Gaines BR (eds) Discrete Systems and Fuzzy Reasoning (Workshop Proc.). Queen Mary College Univ. London, London, pp 110–124
18. Kohout LJ, Pinkava V (1980) The algebraic structure of the Spencer–Brown and Varela calculus. Internat J General Syst 6(3):155–171
19. Krajíček J (1995) Bounded arithmetic, propositional logic, and complexity theory. Cambridge Univ. Press, Cambridge
20. Manber U (1989) Introduction to algorithms. Addison-Wesley, Reading, MA
21. Pinkava V (1966) Logické modely sexualních deviací v objektu (Models of sexual deviations based on many-valued logics). PhD Thesis Charles Univ. Prague
22. Pinkava V (1971) A logical model of some schizophrenic thought impairments. Internat J Man-Machine Studies 3:81–97
23. Pinkava V (1971) Logical models of sexual deviations. Internat J Man-Machine Studies 3:351–374
24. Pinkava V (1971) On a set of functionally complete systems of functors for k-valued logical calculi. Preprint Severalls Hospital, Colchester, UK
25. Pinkava V (1974) A family of complete sets of functions in k-valued logics allowing easy generating. IEEE Comput Repos R74:1–24
26. Pinkava V (1975) Some further properties of the Pi-logics. Proc. 1975 Internat. Symp. Multiple-Valued Logic. IEEE, New York20–26
27. Pinkava V (1976) Arrangements of formulas and minimisation in Pi-algebras. In: Mamdani EH, Gaines BR (eds) Discrete Systems and Fuzzy Reasoning (Workshop Proc.). Queen Mary College Univ. London, London
28. Pinkava V (1976) Fuzzification of binary and finite multivalued logical calculi. Internat J Man-Machine Studies 8:717–730
29. Pinkava V (1977) On the nature of some logical paradoxes. Internat J Man-Machine Studies 9:383–398
30. Pinkava V (1978) On a class of functionally complete multivalued logical calculi. Studia Logica 2:201–212
31. Pinkava V (1978) Variable structure automata. In: Internat. Symp. General Systems Res., Cleveland State Univ., pp 1–26
32. Pinkava V (1979) On Sheffer functions in k-valued logical calculi. Colloq Math Soc János Bolyai, vol 28. North-Holland, Amsterdam, pp 537–545
33. Pinkava V (1979) On some manipulative properties of PI-algebras. In: Proc. Ninth Internat. Symp. Multiple-Valued Logic (Bath, England, 29-31 May 1979). IEEE, New York, pp 139–142
34. Pinkava V (1980) A multivalued logical net modelling conditioning. Behavioral and Brain Sci 3:461–462. Open Peer Commentary on Eysinck HJ (1979) The conditioning model of neurosis, ibid 2:459–482.
35. Pinkava V (1980) On potential tautologies in k-valued calculi. In: Wang PP (ed) Fuzzy Sets: Theory and Applications to Policy Analysis and Information systems. Plenum, New York, pp 77–86
36. Pinkava V (1981) The dichotomous predicament of contemporary psychology. Behavioral and Brain Sci 4:546–547. Open Peer Commentary on Colby KM: Modeling a paranoid mind, 534–550.
37. Pinkava V (1986) Classification and diagnostics. In: Kohout LJ, Bandler W (eds) Knowledge Representation in Medicine and Clinical Behavioural Sci. Abacus Book. Gordon and Breach, New York, pp 69–94
38. Pinkava V (1988) Introduction to logic for system modelling. Gordon and Breach, New York
39. Pinkava V (1993) Multiple-valued logics and classification criteria. J Fuzzy Logic and Intelligent Systems 3(1):90–96,

special issue: Festschrift in Wyllis Bandler's Honor, Zadeh LA, et al (eds)

40. Pitassi T (1997) Algebrais propositional proof systems. In: Immerman N, Kolaitis PG (eds) Descriptive Complexity and Finite Models. Amer. Math. Soc., Providence, RI, pp 215–244
41. Rosenberg I (1965) La structure des functions de plusieurs variables sur un ensemble fini. C R Acad Sci Paris 260:3817–3819
42. Rosenberg I (1970) Über die functionale Vollstandigkeit in den mehrvertigen Logiken. Rozpravy ČSAV, Natural Sci Ser (Prague) 80(4):1–93
43. Salomaa A (1959) On many-valued systems of logic. Ajatus 22:115–159
44. Schweizer B, Sklar A (1983) Probabilistic metric spaces. North-Holland, Amsterdam
45. Slupecki J (1939) A criterion of completness of many-valued systems of propositional logic. CR Soc Sci Letters Varsovie 32:102–110. (In Polish.)

First Order Constraint Qualifications

DIETHARD KLATTE
Institute OR, University Zurich, Zurich, Switzerland

MSC2000: 90C30, 49K27, 90C31, 49K40

Article Outline

Keywords
Optimality Conditions
Duality
Stability
Metric Regularity
Error Bounds
See also
References

Keywords

Constraint qualification; Optimality conditions; Duality; Regularity; Stability

A *constraint qualification* (CQ) is a condition imposed on the analytical description of a given set which can be the constraint set of an optimization problem. CQs are essential in order to establish optimality conditions, but they play also a crucial role in duality theory and perturbation analysis for optimization problems, and in the study of error bounds and stability for algebraic systems like systems of equations or/and inequalities. The notion *first order constraint qualification* is used if a CQ is formulated in terms of first order derivatives or generalized derivatives of the data functions defining the (constraint) set, or if it is related to optimality or stability conditions involving first order terms of the original data. Roughly speaking, first order constraint qualifications establish a link between the geometry of the given set and certain kinds of first order approximations of the analytical data.

A canonical form of constraints for which constraint qualifications have been studied is, for example, the constraint system of a mathematical programming problem, i. e.,

$$\begin{cases} g_i(x) \leq 0, & i \in I = \{1, \dots, m\}, \\ g_j(x) = 0, & j \in J = \{m+1, \dots, r\}, \end{cases} \tag{1}$$

where $g_i : \mathbf{R}^n \to \mathbf{R}$ $(i = 1, \dots, r)$ are given functions, possibly restricted to some subset $X \subset \mathbf{R}^n$.

Another canonical form are *abstract constraints*,

$$G(x) \in C, \tag{2}$$

where G maps a Banach space X into a Banach space Y, and C is a nonempty closed convex cone in Y. Many of the results reported below similarly hold (with some technical modifications) under the weaker assumption that $C \subset Y$ is an arbitrary closed convex set [4,5,8,40,42]. The inclusion (2) is suitable to represent also constraints of abstract optimal control problems, semi-infinite programs, semidefinite optimization problems, and others, see, e. g., [5]. Obviously, (1) is a special case of (2), put $X = \mathbf{R}^n$, $Y = \mathbf{R}^r$, $C = \{y : y_i \leq 0, i \in I; y_j = 0, j \in J\}$ and $G = (g_1, \dots, g_r)$.

The notion 'constraint qualification' was introduced by H.W. Kuhn and A.W. Tucker [22] in developing the theory of nonlinear programming. However, under the name *regularity conditions*, description-depending assumptions were known already in classical variational and extremum problems. To illustrate the meaning of first order CQs, let us consider a simple example:

Example 1

$$\begin{cases} \min & \frac{1}{2}x^2 + y \\ \text{s.t.} & x - y = 0. \end{cases}$$

The classical Lagrange conditions $x + u = 0$, $1 - u = 0$, $x - y = 0$ are necessary (and sufficient, in this example) for the optimality of the point $(\overline{x}, \overline{y}) = (-1, -1)$ with associated multiplier $\overline{u} = 1$. On the other hand, if the constraint is equivalently written as

$$\frac{1}{2}(x - y)^2 = 0,$$

then the corresponding Lagrange conditions become $x + u(x - y) = 0$, $1 - u(x - y) = 0$, $(x - y)^2/2 = 0$, which are contradictory. Trivially, in the first description of the feasible set, the linearization adequately represents the possibilities for variation near $(\overline{x}, \overline{y})$, in the second description, the linearization is inadequate in this respect.

Optimality Conditions

First order necessary optimality conditions in dual form require certain CQs to hold. Consider the optimization problem

$$\text{(P)} \quad \begin{cases} \min & f(x) \\ \text{s.t.} & x \in M, \end{cases}$$

where M is the solution set of (1). First suppose that $X = \mathbf{R}^n$, $f\colon \mathbf{R}^n \to \mathbf{R}$, and g_i $(\forall i)$ are continuously differentiable. For $\overline{x} \in M$ define $I_{\overline{x}} := \{i \in I\colon g_i(\overline{x}) = 0\}$, write $h \in T_M(\overline{x})$ (tangent cone) if $h = \lim_{k\to\infty} \lambda_k(x^k - \overline{x}$, where $\lambda_k > 0$, $x^k \in M$ $(\forall k)$ and $x^k \to \overline{x}$, and write $h \in K_M(\overline{x})$ (linearization cone) if $\langle h, Dg_i(\overline{x})\rangle \leq 0$ for $i \in I_{\overline{x}}$ and $\langle h, Dg_j(\overline{x})\rangle = 0$ for $j \in J$.

Then the *Karush–Kuhn–Tucker conditions* (KKT),

$$\begin{cases} \exists u \in \mathbb{R}^r : & Df(\overline{x}) + \sum\limits_{i\in I\cup J} u_i Dg_i(\overline{x}) = 0, \\ & \overline{x} \in M, \\ & u_i \geq 0, \quad u_i g_i(\overline{x}) = 0, \quad i \in I, \end{cases}$$

are necessary for $\overline{x}$ being a local minimizer of (P), provided that, for example, one of the following CQs is satisfied (see, e. g., [2]):

- *Abadie CQ*: $T_M(\overline{x}) = K_M(\overline{x})$.
- *Kuhn–Tucker CQ*: For every $h \in K_M(\overline{x})$ there is a continuously differentiable function $y\colon [0, \delta) \to M$, $\delta > 0$, such that $y(0) = \overline{x}$ and $\dot{y}(0) = h$.
- *Mangasarian–Fromovitz CQ* (MFCQ, [28]): $Dg_j(\overline{x})$, $j \in J$, are linearly independent, and for some $h \neq 0$, there holds $\langle h, Dg_i(\overline{x})\rangle < 0$, $i \in I_{\overline{x}}$, and $\langle h, Dg_j(\overline{x})\rangle = 0, j \in J$.
- *Linear Independence CQ* (LICQ): $Dg_i(\overline{x})$, $i \in I \cup J$, are linearly independent.

There holds (see, e. g., [2]): LICQ ⇒ MFCQ ⇒ Kuhn–Tucker CQ ⇒ Abadie CQ; the converse implications are not true, in general. For further CQs in this respect, see [2,38]. If no inequalities appear (i. e., $I = \emptyset$), the above CQs are classical for optimality conditions in Euler–Lagrange form. Note that Abadie's CQ is automatically satisfied at each point of M if g_i are affine-linear for all indices $i \in I \cup J$.

Now suppose that (P) is a convex program, i. e., g_i, $i \in I$, are convex (but not necessarily differentiable) functions and g_j are affine-linear functions with gradients a_j, $j \in J$. Then the following CQs are often used for optimality conditions of Karush–Kuhn–Tucker type (in subdifferential terms) and saddle-point conditions ([16,36,41]):

- *Basic CQ* at $\overline{x} \in M$: Each normal direction h, i. e., $\langle h, x - \overline{x}\rangle \leq 0$ for all $x \in M$, has a representation $h = \sum_{i\in I_{\overline{x}}} \mu_i d_i + \sum_{j\in J} \lambda_j a_j$ for some $\lambda \in \mathbf{R}^m$, $\mu_i \geq 0$, $d_i \in \partial g_i(\overline{x})$ (for $i \in I_{\overline{x}}$), where $\partial g_i(\overline{x})$ denotes the *Moreau–Rockafellar subdifferential* of g_i at $\overline{x}$.
- *Weak Slater CQ*: $\exists x^0 \in M$ such that $g_i(x^0) < 0$, $i \in I_N$, where $I_N := \{i \in I\colon g_i$ is not affine-linear$\}$.
- *Strong Slater CQ*: $\exists x^0 \in M$ such that $g_i(x^0) < 0$, $i \in I$, is satisfied, and a_j, $j \in J$, are linearly independent.

The latter naming of CQs is taken from [16]. If no equations appear, the strong Slater CQ becomes the well-known and classical Slater CQ [2,36,41]. There holds: weak Slater CQ ⇒ basic CQ; and for a given $\overline{x} \in M$, the basic CQ is equivalent to a nonsmooth form of the Abadie CQ [16]. If the g_i, $i \in I$, are differentiable, then the strong Slater CQ is equivalent to the MFCQ being satisfied at any $x \in M$ [33,34]. There are certain forms of first order optimality conditions which do not require a CQ, see, e. g., [2,3,32,38].

Next, consider

$$(\widehat{\text{P}}) \quad \begin{cases} \min & f(x) \\ \text{s.t.} & x \in M, \end{cases}$$

where M is the solution set of (2), and f is defined on the Banach space X. Let f, G be continuously differentiable.

Denote by Y^* the dual space of Y. Then the conditions

$$\begin{cases} \exists u \in Y^* : & Df(\overline{x}) + \langle u, DG(\overline{x})\rangle = 0, \\ & \langle u, y\rangle \leq 0, \quad \forall y \in C, \\ & G(\overline{x}) \in C, \\ & \langle u, G(\overline{x})\rangle = 0, \end{cases}$$

are necessary for $\overline{x}$ being a local minimizer of $(\widehat{\mathrm{P}})$, provided that, for example, the following CQ is satisfied (see, e. g., [5,34,42]):

- *Robinson CQ*: $0 \in \operatorname{int}\{G(\overline{x}) + DG(\overline{x})X - C\}$, where 'int' denotes the topological interior.

Because the core of a convex set includes its interior, $0 \in \operatorname{core}\{G(\overline{x}) + DG(\overline{x})X - C\}$ is a consequence of the Robinson CQ. In fact, the latter is also sufficient for Robinson's CQ to hold, and both CQs are also equivalent to $\mathbb{R}_+[G(\overline{x}) + DG(\overline{x})X - C] = Y$, for details one may consult [5,33,42]. If $(\widehat{\mathrm{P}})$ is specialized to (P), then the Robinson CQ and MFCQ are equivalent [34]. Under convexity assumptions on f and G in $(\widehat{\mathrm{P}})$, an extension of the strong Slater CQ plays a crucial role for first order optimality characterizations [37] (see also [40]): $0 \in \operatorname{int}(G(x) - C)$, which becomes $G(x) \in \operatorname{int}C$ if $\operatorname{int}C \neq \emptyset$. In the case of differentiable data, the latter CQ is equivalent to the Robinson CQ [33,40].

For many other classes of optimization problems, first order CQs in connection with optimality conditions have been intensively studied. Among them we refer to CQs in composite optimization [38], optimal control problems [7,17,31], nonsmooth (nonconvex) programs [7,38], mathematical programs with equilibrium constraints [27], semidefinite programs [39], and semi-infinite programs [5,15,31,32]. Certain first order CQs, in particular, Robinson's CQ and the MFCQ play an important role in the theory of second order optimality conditions (and second order stability analysis), see ▶ Second order constraint qualifications and, e. g., [3,4,8,39,40].

Duality

If (P) is a convex program, then first order CQs are closely related to the existence of optimal solutions of the Lagrange dual problem (D) associated with (P) and to properties of the perturbation function $v(u) := \inf\{f(x) \colon g_i(x) \leq u_i,\ i \in I,\ g_j(x) = u_j,\ j \in J\}$, like continuity or subdifferentiability [10,36,37]. An important CQ is

- *Calmness*: $v(0)$ is finite and the Moreau–Rockafellar subdifferential $\partial v(0)$ of $v(\cdot)$ is nonempty.

Under calmness, the dual problem (D) is solvable and $v(0)$ coincides with the optimal value of (D) [10,36,37]. The strong Slater CQ implies calmness. If $v(0)$ is finite, then the following three conditions are mutually equivalent:

i) For (1) the strong Slater CQ holds;

ii) $v(\cdot)$ is continuous at 0;

iii) the set of solutions of the dual problem (D) is nonempty and bounded.

For more details see, e. g., [1,33,36]; for generalizations to convex problems $(\widehat{\mathrm{P}})$ with abstract constraints of the type (2) see, e. g., [33,37,40].

Now suppose that (P) has continuously differentiable data f, g_i, and $\overline{x}$ is a stationary solution solution of (D), i. e., $\overline{x}$ satisfies together with some multiplier u the KKT condition. Then, obviously, LICQ implies that the multiplier u associated with $\overline{x}$ is unique. In [25] is shown that a strengthened form of MFCQ, the so-called strict MFCQ, is necessary and sufficient for the uniqueness of the Lagrange multiplier. Another basic result is the following: MFCQ holds at $\overline{x}$ if and only if the set of all multipliers associated with $\overline{x}$ is bounded (*Gauvin's theorem* [13]).

Extensions of Gauvin's theorem to the general problem $(\widehat{\mathrm{P}})$ with smooth data can be found, e. g., in [8,34,42]. For recent surveys of several aspects of CQs and duality, see [5,40].

The above relations are also important theoretical tools for establishing solution techniques which use Lagrangians or dual schemes (see, e. g., [16,29,38]), for convergence analysis of path following methods (see, e. g., [14]), for regularity properties guaranteeing finite termination of algorithms [6,11], and for several stability subjects (see the next section).

Stability

If the data couples (f, g) of (P) or (f, G) of $(\widehat{\mathrm{P}})$, respectively, are embedded in a family $\mathbf{F}$ of data, where a 'distance' between two elements of $\mathbf{F}$ should be available, then the question arises how changes of the data in $\mathbf{F}$ affect existence of solutions (local or global optimizers, stationary solutions, critical points), and whether 'small' data perturbations lead to 'small' changes of the optimal value and solution set, or not. A 'good' stability behavior is often sensitive to the description of the constraint set and needs a CQ.

For example, consider a parametric smooth program with finite-dimensional variables x and canonical perturbations (t, a, b) in a finite-dimensional space, namely,

$$(\mathrm{P}(t,a,b)) \quad \begin{cases} \min & f(t,x) - \langle a, x\rangle \\ \text{s.t.} & g_i(t,x) \leq b_i, \quad i \in I, \\ & g_j(t,x) = b_j, \quad j \in J, \end{cases} \tag{3}$$

with respect to x, where I, J are as above and f, g_i are twice continuously differentiable with respect to (t, x). Given an initial parameter triple $(\bar{t}, 0, 0)$ and a KKT point $\bar{z} = (\bar{x}, \bar{u})$ of the initial problem, then strong stability of $\bar{z}$ (i. e., the existence of a locally unique and Lipschitzian solution $z(t, a, b)$ of the perturbed KKT system near $\bar{z}$) necessarily requires LICQ to hold at $\bar{x}$ [23], while LICQ together with some strong second order optimality condition characterizes strong stability [9,21,23,35].

MFCQ and the strong Slater CQ are very important to get other stability properties like strong stability, pseudoregularity, upper Lipschitz (or Hölder) continuity, or upper semicontinuity of the optimal and/or stationary solution maps under perturbations (see also the next section). LICQ and MFCQ, respectively, play an essential role for existence, representations and estimates of directional derivatives (studied in different forms: standard one-sided directional derivative, semiderivative, Dini type, Hadamard type, and others) of the optimal value function. For an introduction into these interrelations, see [1,5,9,12,19,24,38], while [5,40] also survey extensions to the class $(\widehat{\mathrm{P}})$, under the Robinson CQ.

In the study of structural (or global) stability of feasible sets and nonlinear finite/semi-infinite programs, including one-parametric deformations, MFCQ and its extensions turn out to be fundamental in these settings, in particular, they characterize the global stability of compact feasible sets, for surveys see [14,18].

If one is interested in directional stability of optimal values or optimal solutions under data perturbations, another type of CQs often comes into play: directional regularity conditions which are imposed on the constraint set of the initial problem $\mathrm{P}((\bar{t}, 0, 0))$. A typical example is the

- *Gollan CQ* at a feasible $\bar{x}$ in direction d: $Dg_j(\bar{t}, \bar{x})$, $j \in J$, are linearly independent, and for some $h \neq 0$, there holds $\langle (h, d), Dg_i(\bar{t}, \bar{x})\rangle < 0$, $i \in I_{(\bar{t}, \bar{x})}$, and $\langle (h, d), Dg_j(\bar{t}, \bar{x})\rangle = 0$, $j \in J$.

For this CQ and a natural extension to abstract constraints in Banach spaces, see [4,5,40], in which directional differentiability and second order expansion of the optimal-value function as well as Lipschitz/Hölder stability and first order expansion of optimal solutions under directional regularity conditions are studied.

Metric Regularity

Metric regularity of a parametric constraint system refers to a certain local error bound for the distance of some point x to the solution set in terms of the residuum of the data at x and is closely related to first order CQs. Consider for example system (1) with $X = \mathbf{R}^n$ under right-hand side perturbations b,

$$\begin{cases} g_i(x) \leq b_i, & i \in I, \\ g_j(x) = b_j, & j \in J, \end{cases} \tag{3}$$

where I and J are as above. Denote by $S(b)$ the solution set mapping of this system. If g_i, $\forall i \in I$, are continuously differentiable, and if $\bar{x} \in S(0)$, then MFCQ at $\bar{x}$ implies that (4) is *metrically regular* at $(\bar{x}, 0)$, i. e., there exist a neighborhood U of $(\bar{x}, 0)$ and a constant $L = L(\bar{x})$ such that for $(x, b) \in U$,

$$\operatorname{dist}(x, S(b)) \leq L \left\| \begin{pmatrix} (g_i(x) - b_i)_+, & i \in I \\ g_j(x) - b_j, & j \in J \end{pmatrix} \right\|, \tag{5}$$

where $c_+ := \max\{c, 0\}$ for $c \in \mathbf{R}$, and $\|\cdot\|$ is any norm in $\mathbf{R}^r$. This was shown in [34], the converse assertion is also true [8,34]. In the Banach space context of the system (2) with right-hand side perturbations, the same equivalence holds when replacing MFCQ by the Robinson CQ, see again [8,34].

If g_i, $i \in I$, are convex (not necessarily differentiable), and g_j, $j \in J$, are affine-linear, then the strong Slater CQ implies that (4) is metrically regular at each $(\bar{x}, 0)$, $\bar{x} \in S(0)$, see [33]. The converse direction is also true [8]. In fact, in both [8] and [33], the authors prove these results for convex inclusions in the Banach space setting (2) using a suitable generalized form of the Slater CQ. Moreover, under the strong Slater CQ, the estimate

(5) even holds for all x in X and all b near 0, where the number $L = L(\overline{x})$ is bounded on bounded sets [33].

Note that under mild assumptions on the parameter-dependence, the same CQs imply metric regularity under more general perturbations (like in (3), for example), for details see again [8,24,33,34].

Error Bounds

The role of (first order) CQs for deriving local and global error bounds will be now discussed for a system of convex inequalities, i. e., suppose in (1) that $J = \emptyset$, and g_i, $i \in I$, are convex (not necessarily differentiable) functions. Denote the solution set again by M.

Given $\overline{x} \in M$, the system (1) is called *metrically regular** at $\overline{x}$ (or simply *metrically regular* at $\overline{x}$ [26,30]; the asterisk is used to avoid confusions with the above notion for parametric systems), if there exist a neighborhood U of $\overline{x}$ and a constant $L = L(\overline{x})$ such that for $x \in U$,

$$\operatorname{dist}(x, M) \leq L \max\{g_i(x)_+ : \ i \in I\}\,. \qquad (6)$$

In [26] was shown that for differentiable functions g_i, $i \in I$, the Abadie CQ holds at $\overline{x} \in M$ if and only if (1) is metrically regular* at $\overline{x}$ [26]. For the nondifferentiable case, it follows by a similar idea of proof that the basic CQ is equivalent to metric regularity*.

If M is bounded and the strong Slater CQ holds, then (6) is satisfied for all $x \in \mathbf{R}^n$, with some uniform constant L [33]. This property is called a *global error bound*, or, an 'error bound in Hoffman's sense' [20,26,30]. If M is unbounded then additional *asymptotic CQs* are required to guarantee the existence of a global error bound. For a survey of asymptotic CQs and their interrelations, see [20].

CQs like Abadie's CQ, MFCQ and the (strong) Slater CQ are also essential in deriving local and global error bounds for approximate solutions of convex and nonconvex mathematical programs and other variational problems. These questions are for some classes of programs closely related to the existence of so-called weak sharp minima introduced in [6,11]. For a general survey of error bounds in the sense just discussed see [30], for the special case of quadratic convex programs see [26].

In contrast to other applications of CQs, the relations between CQs and error bounds are still not clarified completely and require strong additional effort in research.

See also

- ▶ Equality-constrained Nonlinear Programming: KKT Necessary Optimality Conditions
- ▶ Inequality-constrained Nonlinear Optimization
- ▶ Kuhn–Tucker Optimality Conditions
- ▶ Lagrangian Duality: Basics
- ▶ Rosen's Method, Global Convergence, and Powell's Conjecture
- ▶ Saddle Point Theory and Optimality Conditions
- ▶ Second Order Constraint Qualifications
- ▶ Second Order Optimality Conditions for Nonlinear Optimization

References

1. Bank B, Guddat J, Klatte D, Kummer B, Tammer K (1982) Non-linear parametric optimization. Akad. Verlag, Berlin
2. Bazaraa MS, Sherali HD, Shetty CM (1993) Nonlinear programming: Theory and algorithms, 2nd edn. Wiley, New York
3. Ben-Israel A, Ben-Tal A, Zlobec S (1981) Optimality in nonlinear programming: A feasible direction approach. Wiley, New York
4. Bonnans JF, Cominetti R (1996) Perturbed optimization in Banach spaces I, II, III. SIAM J Control Optim 34:1151–1171; 1172–1189; 1555–1567
5. Bonnans JF, Shapiro A (2000) Perturbation analysis of optimization problems. Springer, Berlin
6. Burke JV, Ferris MC (1993) Weak sharp minima in mathematical programming. SIAM J Control Optim 31:1340–1359
7. Clarke F (1983) Optimization and nonsmooth analysis. Wiley, New York
8. Cominetti R (1990) Metric regularity, tangent sets and second-order optimality conditions. Appl Math Optim 21:265–287
9. Dontchev A, Rockafellar RT (1997) Characterizations of Lipschitz stability in nonlinear programming. In: Fiacco AV (ed) Mathematical Programming with Data Perturbations. M. Dekker, New York, pp 65–82
10. Ekeland I, Temam R (1974) Analyse convexe et problèmes variationnels. Dunod, Paris
11. Ferris MC, Mangasarian OL (1992) Minimum principle sufficiency. Math Program 57:1–14
12. Fiacco AV (1983) Introduction to sensitivity and stability analysis. Acad. Press, New York
13. Gauvin J (1977) A necessary and sufficient regularity condition to have bounded multipliers in nonconvex programming. Math Program 12:136–138

14. Guddat J, Guerra F, Jongen HTh (1990) Parametric optimization: Singularities, pathfollowing and jumps. Wiley, New York
15. Hettich R, Kortanek K (1993) Semi-infinite programming: Theory, methods, and applications. SIAM Rev 35:380–429
16. Hiriart-Urruty J-B, Lemaréchal C (1993) Convex analysis and minimization algorithms I, II. Springer, Berlin
17. Ioffe AD, Tichomirov VM (1974) Theory of extremal problems. Nauka, Moscow
18. Jongen HTh, Rückmann Jan-J (1998) On stability and deformation in semi-infinite optimization. In: Reemtsen R, Rückmann J-J (eds) Semi-Infinite Programming. Kluwer, Dordrecht, pp 29–67
19. Klatte D (1994) On quantitative stability for non-isolated minima. Control and Cybernetics 23:183–200
20. Klatte D, Li W (1999) Asymptotic constraint qualifications and global error bounds for convex inequalities. Math Program 84:137–160
21. Kojima M (1980) Strongly stable stationary solutions on nonlinear programs. In: Robinson SM (ed) Analysis and Computation of Fixed Points. Acad. Press, New York, pp 93–138
22. Kuhn HW, Tucker AW (1950) Nonlinear programming. In: Neyman J (ed) Proc. Second Berkeley Symp. Math. Stat. Probab. Univ. Calif. Press, Berkeley, CA, pp 481–492
23. Kummer B (1991) Lipschitzian inverse functions, directional derivatives and application in $C^{1,1}$ optimization. J Optim Th Appl 70:559–580
24. Kummer B (1997) Lipschitzian and pseudo-Lipschitzian inverse functions and applications to nonlinear programming. In: Fiacco AV (eds) Mathematical Programming with Data Perturbations. M. Dekker, New York, pp 201–222
25. Kyparisis J (1985) On uniqueness of Kuhn–Tucker multipliers in nonlinear programming. Math Program 32:242–246
26. Li W (1997) Abadie's constraint qualification, metric regularity, and error bounds for differentiable convex inequalities. SIAM J Optim 7:966–978
27. Luo Z-Q, Pang J-S, Ralph D (1996) Mathematical programs with equlibrium constraints. Cambridge Univ. Press, Cambridge
28. Mangasarian OL, Fromovitz S (1967) The Fritz John necessary optimality conditions in the presence of equality and inequality constraints. J Math Anal Appl 17:37–47
29. Nesterov Y, Nemirovskii A (1994) Interior-point polynomial algorithms in convex programming. SIAM Stud Appl Math. SIAM, Philadelphia
30. Pang J-S (1997) Error bounds in mathematical programming. Math Program 79:299–332
31. Polak E (1997) Optimization: Algorithms and consistent approximations. Springer, Berlin
32. Pshenichnyi BN (1969) Necessary conditions for an extremum. Nauka, Moscow
33. Robinson SM (1976) Regularity and stability for convex multivalued functions. Math Oper Res 1:130–143
34. Robinson SM (1976) Stability theorems for systems of inequalities. Part II: Differentiable nonlinear systems. SIAM J Numer Anal 13:497–513
35. Robinson SM (1980) Strongly regular generalized equations. Math Oper Res 5:43–62
36. Rockafellar RT (1970) Convex analysis. Princeton Univ. Press, Princeton
37. Rockafellar RT (1974) Conjugate duality and optimization. Regional Conf Ser Applied Mat. SIAM, Philadelphia
38. Rockafellar RT, Wets RJ-B (1997) Variational analysis. Springer, Berlin
39. Shapiro A (1997) First and second order analysis of nonlinear semidefinite programs. Math Program 77:301–320
40. Shapiro A (1998) First and second order optimality conditions and perturbation analysis of semi-infinite programming problems. In: Reemtsen R, Rückmann J-J (eds) Semi-Infinite Programming. Kluwer, Dordrecht, pp 103–133
41. Stoer J, Witzgall C (1970) Convexity and optimization in finite dimensions I. Springer, Berlin
42. Zowe J, Kurcyusz S (1979) Regularity and stability for the mathematical programming problem in Banach spaces. Appl Math Optim 5:42–62

Flow Shop Scheduling Problem

MAGDALENE MARINAKI
Department of Production Engineering
and Management, Industrial Systems
Control Laboratory, Technical University of Crete,
Chania, Greece

MSC2000: 68M20, 90B35

Article Outline

Introduction
Variations
Exact Algorithms for the Flow Shop Scheduling Problem
Heuristic Algorithms for the Flow Shop Scheduling Problem
Metaheuristic Algorithms for the Flow Shop Scheduling Problem
References

Introduction

The general flow-shop problem [4,60,68], denoted as $n/m/C_{\max}$ in the literature, involves n jobs, each requiring operations on m machines, in the same machine sequence. The processing time for each operation is p_{ij}, where $i \in \{1, 2, \dots, n\}$ denotes a job and $j \in \{1, 2, \dots, m\}$ a machine. The problem is to determine the sequence of these n jobs that produces the smallest makespan assuming no preemption of operations. In the simplest case, all jobs are available and ready to start at time zero and the setup times on machines are assumed to be sequence-independent and included in p_{ij}. In more realistic situations, however, jobs are released at different times, thus requiring dynamic scheduling and the setup times are sequence-dependent.

The makespan problem for flow shops has been the most studied by far in the literature. (The makespan $C_{\max}$ is equivalent to the completion time of the last job to leave the system.) This is partly because:

- Makespan is a simple and useful criterion for heavily loaded shops when long-term utilization should be maximized.
- Makespan is the only objective function simple enough to have available some analytic results for multimachine problems and to make some branch-and-bound methods practical for medium-sized problems.

The minimization of the makespan objective is to a certain extent equivalent to the maximization of the utilization of the machines. The models, however, tend to be of such complexity that makespan results are already relatively hard to obtain. Even harder to analyze are the flow time and the due-date-related objectives.

Variations

There are a number of variations for the flow shop scheduling problem [60,68]. Some of them are presented in the following.

The **permutation flow shop problem** (PFSP). A constraint that may appear in the flow-shop environment is that the queues in front of each machine operate according to the first in, first out discipline, which implies that the order (or permutation) in which the jobs go through the first machine is maintained throughout the system. This problem can be formulated as follows. Each of n jobs from the job set $j = \{1, 2, \dots, n\}$, for $n > 1$, has to be processed on m machines $1, 2, \dots, m$ in the order given by the indexing of the machines. Thus, job $j, j \in J$, consists of a sequence of m operations; each of them corresponding to the processing of job j on machine i during an uninterrupted processing time $p_{ij} \geq 0$. (It is assumed that a zero processing time on a machine corresponds to a job performed by that machine in an infinitesimal time.) Machine $i, i = 1, 2, \dots, m$, can execute at most one job at a time, and it is assumed that each machine processes the jobs in the same order. We represent the job processing order by the permutation $\pi = (\pi(1), \dots, \pi(n))$ on the set j, and we let P denote the set of all permutations on j. We wish to find the optimal processing order $\pi^* \in P$ of jobs minimizing the maximum completion time $C_{\max}(\pi)$ (makespan) [65,68].

The **flow shop scheduling problem with limited machine availability**. In such a problem, n jobs have to be scheduled on m machines under the makespan criterion and under the assumption that the machines are not available during the whole planning horizon [6].

No-wait or no-idle flow shop scheduling problems with deteriorating jobs. Deterioration of a job means that its processing time is a function of its execution start time. A simple linear deterioration function is assumed and some dominating relationships between machines can be satisfied. No-wait requirement is another phenomenon which may occur in flow shops and implies that the starting time of a job at the first machine has to be delayed to ensure that the job can go through the flow shop without having to wait for any machine. The "no-idle" constraint means that each machine, once it commences its work, has to process all operations assigned to it without any interruption. In [102] it is shown that for the problems to minimize makespan or weighted sum of completion time, polynomial algorithms still exist, although these problems are more complicated than the classical ones. In [101] the general, no-wait and no-idle flow shop scheduling problem with decreasing linear deterioration under dominant machines is considered.

A **hybrid flow shop** consists of a series of production stages, each of which has several machines operating in parallel. Some stages may have only one machine, but at least one stage must have multiple machines. The flow of jobs through the shop is unidirectional. Each job is processed by one machine in each stage and it must go through one or more stage. Machines in each stage can be identical, uniform or unrelated. An extended survey of the problem is presented in [57].

In [45] the **stochastic flow shop problem** is presented and analyzed.

Exact Algorithms for the Flow Shop Scheduling Problem

Branch and bound is a general method for solving many types of combinatorial problems. The basic idea of branching is to conceptualize the problem as a decision tree. From each decision choice point, called a node, for a partially completed solution there grow a number of new branches, one for each possible decision. These in turn become new nodes for branching again and so on. Leaf nodes, which cannot branch any further, represent complete solutions or dead ends. A number of branch-and-bound procedures have been proposed for the solution of the flow shop scheduling problem and its variations [16,19,36,83,95,106,107]. Dynamic programming approaches for the solution of the flow shop scheduling problem have been proposed in [92,105].

Heuristic Algorithms for the Flow Shop Scheduling Problem

Since the last few decades, pure flow shop scheduling problems have been largely studied. Since the flow shop minimization problem is NP-hard [87], a number of heuristic and metaheuristic algorithms have been proposed for the solution of the problem. High-performance heuristics have been proposed to minimize the makespan [15,21,61] or the maximum tardiness [96]. Some additional characteristics have been studied for the makespan criterion: non-sequence-dependent setup and removal times [69,91], minimum time lags [91], and more recently job-precedence constraints [14]. Studies on hybrid flow shop scheduling problems are relatively recent. The main results deal with the makespan criterion, and are often limited to two stages; nevertheless, some work has been done on lateness criteria [38,43]. A number of heuristics algorithms were proposed in [13]. A worst-case analysis of heuristics is presented in [85].

In [6] a heuristic approach is proposed to approximately solve the problem that consists in scheduling the jobs two by two according to an input sequence, and using a polynomial algorithm. This algorithm is an extension of the geometric approach developed for the two-job shop scheduling problem. An algorithm that constructs heuristics that use a lower bound to find a feasible solution for the general m-stage flow shop scheduling problem with multiple operations and time lags is described in [75]. A greedy algorithm for the solution of the permutation flow shop model with variable processing times is presented in [28]. A two-stage heuristic algorithm for the flow-shop problem with multiple processors is presented in [90]. A bilevel programming heuristic is presented in [48]. A simple heuristic is presented in [55].

A two-phase heuristic is presented in [89]. In the first phase, an initial job sequence is generated using one of the available well-known and efficient heuristics, while in the second phase the sequence generated is improved in terms of the makespan using a pair exchange mechanism with directionality constraint. The n-job two-machine flow shop scheduling problem is studied in [99] with the criterion of minimizing the sum of job completion times. The scheduling problem is first formulated mathematically. Three heuristic methods are then invented to find near optimal schedules. Three lower bound generation schemata are designed to compute three different lower bounds, of which the tightest one is used. To further reduce the search space, some dominance properties are proved. Then a branch-and-bound algorithm is developed to obtain an optimal schedule. In [100] the flow shop scheduling problem, with the criterion of minimizing the sum of job completion times is addressed. Two heuristic approaches are proposed to deal with this problem. The first approach focuses on reducing machine idle times and the second one places efforts on reducing both the machine idle times and the job queue times.

Complete reviews of the heuristic and metaheuristic algorithms for the solution of the flow-shop problem and some of its variations are presented in [11,20,39,60,68,78,97].

Metaheuristic Algorithms for the Flow Shop Scheduling Problem

Several metaheuristic algorithms have been proposed for the solution of the flow shop scheduling problem. In the following an analytical presentation of these algorithms is given:

- **Simulated annealing** [1,3,50] plays a special role within local search for two reasons. First, it appears to be quite successful when applied to a broad range of practical problems. Second, some threshold-accepting algorithms such as simulated annealing have a stochastic component, which facilitates a theoretical analysis of their asymptotic convergence. Simulated annealing [2] is a stochastic algorithm that allows random uphill jumps in a controlled fashion in order to provide possible escapes from poor local optima. Gradually the probability allowing the objective function value to increase is lowered until no more transformations are possible. Simulated annealing owes its name to an analogy with the annealing process in condensed-matter physics, where a solid is heated to a maximum temperature at which all particles of the solid randomly arrange themselves in the liquid phase, followed by cooling through careful and slow reduction of the temperature until the liquid is frozen with the particles arranged in a highly structured lattice and minimal system energy. This ground state is reachable only if the maximum temperature is sufficiently high and the cooling sufficiently slow. Otherwise a metastable state is reached. The metastable state is also reached with a process known as quenching, in which the temperature is instantaneously lowered. Its predecessor is the so-called Metropolis filter. Simulated annealing algorithms for the flow shop scheduling problem are presented in [27,35,40,46,47,52,58,62,63,82,88,94,98,103].
- **Tabu search** (TS) was introduced by Glover [30,31] as a general iterative metaheuristic for solving combinatorial optimization problems. Computational experience has shown that TS is a well-established approximation technique, which can compete with almost all known techniques and which, by its flexibility, can beat many classic procedures. It is a form of local neighbor search. Each solution S has an associated set of neighbors $N(S)$. A solution $S' \in N(S)$ can be reached from S by an operation called a *move*. TS can be viewed as an iterative technique which explores a set of problem solutions, by repeatedly making moves from one solution S to another solution S' located in the neighborhood $N(S)$ of S [32]. TS moves from a solution to its best admissible neighbor, even if this causes the objective function to deteriorate. To avoid cycling, solutions that have been recently explored are declared *forbidden or tabu* for a number of iterations. The tabu status of a solution is overridden when certain criteria (*aspiration criteria*) are satisfied. Sometimes, *intensification* and *diversification* strategies are used to improve the search. In the first case, the search is accentuated in the promising regions of the feasible domain. In the second case, an attempt is made to consider solutions in a broad area of the search space. TS algorithms for the flow shop scheduling problem are presented in [5,7,8,12,26,27,37,40,46,63,66,86,104].
- **Genetic algorithms** (GAs) are search procedures based on the mechanics of natural selection and natural genetics. The first GA was developed by John H. Holland [42] in the 1960s to allow computers to evolve solutions to difficult search and combinatorial problems, such as function optimization and machine learning. GAs offer a particularly attractive approach for problems like the flow shop scheduling problem since they are generally quite effective for rapid global search of large, nonlinear and poorly understood spaces. Moreover, GAs are very effective in solving large-scale problems. GAs [34] mimic the evolution process in nature. GAs are based on an imitation of the biological process in which new and better populations among different species are developed during evolution. Thus, unlike most standard heuristics, GAs use information about a population of solutions, called individuals, when they search for better solutions. A GA is a stochastic iterative procedure that maintains the population size constant in each iteration, called a generation. The basic operation is the mating of two solutions in order to form a new solution. To form a new population, a binary operator, called crossover, and a unary operator, called mutation, are applied [72,73]. Crossover takes two individuals, called parents, and produces two new individuals, called offspring, by swapping parts of the parents. GAs for the flow shop scheduling problem are presented in [5,9,10,17,18,53,59,64,79,80,81,82].
- **Scatter search** [33] may be viewed as an evolutionary (population-based) algorithm that constructs solutions by combining others. It derives its foundations from strategies originally proposed for combining decision rules and constraints in the context

of integer programming. The goal of this method is to enable the implementation of solution procedures that can derive new solutions from combined elements in order to yield better solutions than those procedures that base their combinations only on a set of original elements. Scatter search algorithms for the flow shop scheduling problem are presented in [44,65].

- **Variable neighborhood search** is a metaheuristic for solving combinatorial optimization problems whose basic idea is systematic change of the neighborhood within a local search [41]. Variable neighborhood search algorithms for the flow shop scheduling problem are presented in [47,63].
- The use of **artificial neural networks** to find good solutions to combinatorial optimization problems has recently attracted some attention. A neural network consists of a network [67] of elementary nodes (neurons) that are linked through weighted connections. The nodes represent computational units, which are capable of performing a simple computation, consisting of a summation of the weighted inputs, followed by the addition of a constant called the threshold or bias, and the application of a nonlinear response (activation) function. The result of the computation of a unit constitutes its output. This output is used as an input for the nodes to which it is linked through an outgoing connection. The overall task of the network is to achieve a certain network configuration, for instance, a required input–output relation, by means of the collective computation of the nodes. This process is often called *self-organization*. A neural networks algorithm for the flow shop scheduling problem is presented in [86].
- An improvement heuristic based on an **adaptive learning approach** is proposed and applied to the general flow-shop problem. The approach uses a single-pass or a constructive heuristic and tries to find improvements iteratively by perturbing the data using a weight factor, allowing a nondeterministic local neighborhood search. The weights are modified using strategies similar to neural-networks training, i. e., weights are reinforced if the solution improves [4].
- **Artificial immune system** (AIS) is an intelligent problem-solving technique that has been used in scheduling problems for about 10 years. AISs are computational systems inspired by theoretical immunology, observed immune functions, principles and mechanisms in order to solve problems. Nature and in particular biological systems have always been fascinating to the human expert owing to their complexity, flexibility and sophistication. The nervous system inspired the evolution of an artificial neural network, in the very similar manner immune system motivated the emergence of the AIS. The AIS can be defined as an abstract or metamorphic computational system using ideas gleaned from the theories and component of immunology [22,23]. AIS algorithms for the flow shop scheduling problem are presented in [25,51].
- **Particle swarm optimization** (PSO) is a population-based swarm intelligence algorithm. It was originally proposed by Kennedy and Eberhart [49] as a simulation of the social behavior of social organisms such as bird flocking and fish schooling. PSO uses the physical movements of the individuals in the swarm and has a flexible and well-balanced mechanism to enhance and adapt to the global and local exploration abilities. PSO algorithms for the flow shop scheduling problem are presented in [7,8,54,56,93].
- The **ant colony optimization** (ACO) metaheuristic is a relatively new technique for solving combinatorial optimization problems. Based strongly on the ant system metaheuristic developed by Dorigo, Maniezzo and Colorni [24], ant colony optimization is derived from the foraging behavior of real ants in nature. The main idea of ACO is to model the problem as the search for a minimum cost path in a graph. Artificial ants walk through this graph, looking for good paths. Each ant has a rather simple behavior so that it will typically only find rather poor-quality paths on its own. Better paths are found as the emergent result of the global cooperation among ants in the colony. An ACO algorithm consists of a number of cycles (iterations) of solution construction. During each iteration a number of ants (which is a parameter) construct complete solutions using heuristic information and the collected experiences of previous groups of ants. These collected experiences are represented by a digital analogue of trail pheromone which is deposited on the constituent elements of a solution. Small quantities

are deposited during the construction phase, while larger amounts are deposited at the end of each iteration in proportion to solution quality. Pheromone can be deposited on the components and/or the connections used in a solution depending on the problem. ACO algorithms for the flow shop scheduling problem are presented in [29,70,71,84].

- **Greedy randomized adaptive search procedure** (GRASP) [74] is an iterative two-phase search method which has gained considerable popularity in combinatorial optimization. Each iteration consists of two phases, a construction phase and a local search procedure. In the construction phase, a randomized greedy function is used to build up an initial solution. This randomized technique provides a feasible solution within each iteration. This solution is then exposed for improvement attempts in the local search phase. The final result is simply the best solution found over all iterations. GRASP algorithms for the flow shop scheduling problem are presented in [76,77].

References

1. Aarts E, Korst J (1989) Simulated Annealing and Boltzmann Machines – A Stochastic Approach to Combinatorial Optimization and Neural Computing. Wiley, Chichester
2. Aarts E, Korst J, Van Laarhoven P (1997) Simulated Annealing. In: Aarts E, Lenstra JK (eds) Local Search in Combinatorial Optimization. Wiley, Chichester, pp 91–120
3. Aarts E, Ten Eikelder HMM (2002) Simulated Annealing. In: Pardalos PM, Resende MGC (eds) Handbook of Applied Optimization. Oxford University Press, pp 209–221
4. Agarwal A, Colak S, Eryarsoy E (2006) Improvement Heuristic for the Flow-Shop Scheduling Problem: An Adaptive-Learning Approach. Eur J Oper Res 169:801–815
5. Aggoune R (2004) Minimizing the Makespan for the Flow Shop Scheduling Problem with Availability Constraints. Eur J Oper Res 153:534–543
6. Aggoune R, Portmann M-C (2006) Flow Shop Scheduling Problem with Limited Machine Availability: A Heuristic Approach. Int J Prod Econ 99:4–15
7. Al-Anzi FS, Allahverdi A (2007) A Self-Adaptive Differential Evolution Heuristic for Two-Stage Assembly Scheduling Problem to Minimize Maximum Lateness With Setup Times. Eur J Oper Res 182(1):80–94
8. Allahverdi A, Al-Anzi FS (2006) A PSO and a Tabu Search Heuristics for the Assembly Scheduling Problem of the Two-Stage Distributed Database Application. Comput Oper Res 33(4):1056–1080
9. Arroyo JEC, Armentano VA (2005) Genetic Local Search for Multi-Objective Flowshop Scheduling Problems. Eur J Oper Res 167(3):717–738
10. Bagchi TP (1999) Multiobjective Scheduling by Genetic Algorithms. Kluwer, Boston
11. Bagchi TP, Gupta JND, Sriskandarajah C (2006) A Review of TSP Based Approaches for Flowshop Scheduling. Eur J Oper Res 169(3):816–854
12. Ben-Daya M, Al-Fawzan M (1998) A Tabu Search Approach for the Flow Shop Scheduling Problem. Eur J Oper Res 109:88–95
13. Botta-Genoulaz V (2000) Hybrid Flow Shop Scheduling with Precedence Constraints and Time Lags to Minimize Maximum Lateness. Int J Prod Econ 64:101–111
14. Botta V, Guinet A (1996) Scheduling flowshops with precedence constraints and time lags. Proceedings of the Workshop on Production Planning and Control, Mons, Belgium, 16–19
15. Campbell HG, Dudek RA, Smith ML (1970) A Heuristic Algorithm for the n-job, m-Machine Sequencing problem. Manag Sci 16:B630–B637
16. Carlier J, and Rebaï, I (1996) Two Branch and Bound Algorithms for the Permutation Flow Shop Problem. Eur J Oper Res 90:238–251
17. Chang PC, Chen SH, Liu CH (2007) Sub-Population Genetic Algorithm with Mining Gene Structures for Multiobjective Flowshop Scheduling Problems. Exp Syst Appl 33(3):762–771
18. Cheng BW, Chang CL (2007) A Study on Flowshop Scheduling Problem Combining Taguchi Experimental Design and Genetic Algorithm. Exp Syst Appl 32(2):415–421
19. Chung CS, Flynn J, Kirca O (2006) A Branch and Bound Algorithm to Minimize the Total Tardiness for M-Machine Permutation Flowshop Problems. Eur J Oper Res 174(1):1–10
20. Conway RW, Maxwell WL, Miller LW (2003) Theory of Scheduling. Dover Publications INC., Mineola
21. Dannenbring DG (1977) An Evaluation of Flowshop Sequencing Heuristics. Manag Sci 23(11):1174–1182
22. De Castro LN, Von Zuben FJ (1999) Artificial Immune Systems, Part 1, Basic Theory and Applications. Technical Report, TR-DCA 01/99
23. De Castro LN, Von Zuben FJ (2001) Learning and Optimization Using the Clonal Selection Principle. Trans IEEE Evol Comput 6(3):239–251
24. Dorigo M, Maniezzo V, Colorni A (1996) Ant System: Optimization by a colony of cooperating agents. IEEE Trans Syst Man Cybern B 26(1):29–41
25. Engin O, Döyen A (2004) A New Approach to Solve Hybrid Flow Shop Scheduling Problems by Artificial Immune System. Future Generat Comput Syst 20:1083–1095

26. Eren T, Guner E (2006) A Bicriteria Flowshop Scheduling Problem with Setup Times. Appl Math Comput 183(2):1292–1300
27. Fink A, Voßb S (2003) Solving the Continuous Flow-Shop Scheduling Problem by Metaheuristics. Eur J Oper Res 151:400–414
28. Finke G, Jiang H (1997) A Variant of the Permutation Flow Shop Model with Variable Processing Times. Discret Appl Math 76:123–140
29. Gajpal Y, Rajendran C (2006) An Ant-Colony Optimization Algorithm for Minimizing the Completion-Time Variance of Jobs in Flowshops. Int J Prod Econ 101(2):259–272
30. Glover F (1989) Tabu Search I. ORSA J Comput 1(3):190–206
31. Glover F (1990) Tabu Search II. ORSA J Comput 2(1):4–32
32. Glover F, Laguna M, Taillard E, de Werra D (eds) (1993) Tabu Search. JC Baltzer AG, Science Publishers, Basel
33. Glover F, Laguna M, Marti R (2003) Scatter Search and Path Relinking. In: Glover F, Kochenberger GA (eds) Advances and Applications Handbook of Metaheuristics. Kluwer, Boston, pp 1–36
34. Goldberg DE (1989) Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley Publishing Company INC, Massachussets
35. Gourgand M, Grangeon N, Norre S (2003) A Contribution to the Stochastic Flow Shop Scheduling Problem. Eur J Oper Res 151:415–433
36. Gowrishankar K, Rajendran C, Srinivasan G (2001) Flow Shop Scheduling Algorithms for Minimizing the Completion Time Variance and the Sum of Squares of Completion Time Deviations from a Common Due Date. Eur J Oper Res 132:643–665
37. Grabowski J, Pempera J (2005) Some Local Search Algorithms for No-Wait Flow-Shop Problem with Makespan Criterion. Comput Oper Res 32:2197–2212
38. Guinet A, Solomon M (1996) Scheduling Hybrid Flowshops to Minimize Maximum Tardiness or Maximum Completion Time. Int J Prod Res 34(6):1643–1654
39. Gupta JND, Stafford EF (2006) Flowshop Scheduling Research After Five Decades. Eur J Oper Res 169(3):699–711
40. Gupta JND, Henning K, Werner F (2002) Local Search Heuristics for Two-Stage Flow Shop Problems with Secondary Criterion. Comput Oper Res 29:123–149
41. Hansen P, Mladenovic N (2001) Variable Neighborhood Search: Principles and Applications. Eur J Oper Res 130:449–467
42. Holland JH (1975) Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor
43. Hunsucker JL, Shah JR (1992) Performance of priority rules in a due-date flowshop. OMEGA Int J Manag Sci 20(1):73–89
44. Jain AS, Meeran S (2002) A Multi-Level Hybrid Framework Applied to the General Flow-Shop Scheduling Problem. Comput Oper Res 29:1873–1901
45. Jia C (1998) Minimizing Variation in Stochastic Flow Shop. Oper Res Lett 23:109–111
46. Janiak A, Kozan E, Lichtenstein M, Oğuz C (2005) Metaheuristic Approaches to the Hybrid Flow Shop Scheduling Problem with a Cost-Related Criterion. Int J Prod Econ 31(3):504–514
47. Jin Z, Yang Z, Ito T (2006) Metaheuristic Algorithms for the Multistage Hybrid Flowshop Scheduling Problem. Int J Prod Econ 100(2):322–334
48. Karlor JK, Wang W (1996) Bilevel Programming Applied to the Flow Shop Scheduling Problem. Comput Oper Res 23(5):443–451
49. Kennedy J, Eberhart R (1995) Particle Swarm Optimization. In: Proceedings of 1995 IEEE International Conference on Neural Networks 4:1942–1948
50. Kirkpatrick S, Gelatt CD, Vecchi MP (1982) Optimization by Simulated Annealing. Science 220:671–680
51. Kumar A, Prakash A, Shankar R, Tiwari MK (2005) Psycho-Clonal Algorithm Based Approach to Solve Continuous Flow Shop Scheduling Problem. Exp Syst Appl, in press
52. Laha D, Chakraborty UK (2007) An Efficient Stochastic Hybrid Heuristic for Flowshop Scheduling. Eng Appl Artif Intell 20(6):851–856
53. Leu S-S, Hwang S-T (2002) GA-Based Resource-Constrained Flow-Shop Scheduling Model for Mixed Precast Production. Automat Construct 11:439–452
54. Lian Z, Gu X, Jiao B (2006) A Similar Particle Swarm Optimization Algorithm for Permutation Flowshop Scheduling to Minimize Makespan. Appl Math Comput 175(1):773–785
55. Liao C-J, Sun C-L, You W-C (1995) Flow-Shop Scheduling with Flexible Processors. Comput Oper Res 22(3):297–301
56. Liao CJ, Tseng CT, Luarn P (2007) A Discrete Version of Particle Swarm Optimization for Flowshop Scheduling Problems. Comput Oper Res 34(10):3099–3111
57. Linn R, Zhang W (1999) Hybrid Flow Shop Scheduling: A Survey. Comput Indust Eng 37:57–61
58. Low C (2005) Simulated Annealing Heuristic for Flow Shop Scheduling Problems with Unrelated Parallel Machines. Comput Oper Res 32:2013–2025
59. Martin CH (2006) A Hybrid Genetic Algorithm/ Mathematical Programming Approach to the Multi-Family Flowshop Scheduling Problem with Lot Streaming. Omega, In Press, Available online 26 December 2006. doi: 10.1016/j.omega.2006.11.002
60. Morton TE, Pentico DW (1993) Heuristic Scheduling Systems. With Applications to Production Systems and Project Management. Wiley, New York
61. Nawaz M, Enscore E, Ham I (1983) A Heuristic Algorithm for the m-Machine, n-Job Flowshop Sequencing Problem. Omega 11:91–95
62. Nearchou A (2004) A Novel Metaheuristic Approach for the Flow Shop Scheduling Problem. Eng Appl Artif Intell 17:289–300

63. Negenman EG (2001) Local Search Algorithms for the Multiprocessor Flow Shop Scheduling Problem. Eur J Oper Res 128:147–158
64. Neppali VR, Chen C-L, Gupta JND (1996) Genetic Algorithms for the Two-Stage Bicriteria Flow Shop Problem. Eur J Oper Res 95:356–373
65. Nowicki E, Smutnicki C (2006) Some Aspects of Scatter Search in the Flow-Shop Problem. Eur J Oper Res 169:654–666
66. Oğuz C, Zinder Y, Do VH, Janiak A, Lichtenstein M (2004) Hybrid Flow-Shop Scheduling Problems with Multiprocessor Task Systems. Eur J Oper Res 152:115–131
67. Sodererg B, Peterson C (1997) Artificial Neural Networks. In: Aarts E, Lenstra JK (eds) Local Search in Combinatorial Optimization. Wiley, Chichester, pp 173–214
68. Pinedo M (1995) Scheduling. Theory, Algorithms, and Systems. Prentice Hall, Englewood Cliffs
69. Proust C , Gupta JND, Deschamps V (1991) Flowshop Scheduling with Set-Up, Processing and Removal Times Separated. Int J Prod Res 29:479–493
70. Rajendran C, Ziegler H (2004) Ant-Colony Algorithms for Permutation Flowshop Scheduling to Minimize Makespan/Total Flowtime of Jobs. Eur J Oper Res 155(2): 426–438
71. Rajendran C, Ziegler H (2005) Two Ant-Colony Algorithms for Minimizing Total Flowtime in Permutation Flowshops Computers and Industrial Engineering. Comput Indust Eng 48(4):789–797
72. Reeves CR (1995) Genetic Algorithms. In: Reeves CR (ed) Modern Heuristic Techniques for Combinatorial Problems. McGraw-Hill, London, pp 151–196
73. Reeves CR (2003) Genetic Algorithms. In: Glover F, Kochenberger GA (eds) Handbooks of Metaheuristics. Kluwer, Dordrecht, pp 55–82
74. Resende MGC, Ribeiro CC (2003) Greedy Randomized Adaptive Search Procedures. In: Glover F, Kochenberger GA (eds) Handbook of Metaheuristics. Kluwer, Boston, pp 219–249
75. Riezebos J, Gaalman GJC (1998) Time Lag Size in Multiple Operations Flow Shop Scheduling Heuristics. Eur J Oper Res 105:72–90
76. Rios-Mercado R, Bard J (1998) Heuristics for the Flow Line Problem with Setup Costs. Eur J Oper Res 110:76–98
77. Rios-Mercado R, Bard J (1999) An Enhanced TSP-based Heuristic for Makespan Minimization in a Flow Shop with Setup Costs. J Heuristic 5:57–74
78. Ruiz R, Maroto C (2005) A Comprehensive Review and Evaluation of Permutation Flowshop Heuristics. Eur J Oper Res 165(2):479–494
79. Ruiz R, Maroto C (2006) A Genetic Algorithm for Hybrid Flowshops with Sequence Dependent Setup Times and Machine Eligibility. Eur J Oper Res 169(3):781–800
80. Ruiz R, Maroto C, Alcaraz J (2006) Solving the Flowshop Scheduling Problem with Sequence Dependent Setup Times Using Advanced Metaheuristics. Eur J Oper Res 165(1):34–54
81. Ruiz R, Maroto C, Alcaraz J (2006) Two New Robust Genetic Algorithms for the Flowshop Scheduling Problem. Omega 34(5):461–476
82. Sadegheih A (2006) Scheduling Problem Using Genetic Algorithm, Simulated Annealing and the Effects of Parameter Values on GA Performance. Appl Math Modell 30(2):147–154
83. Sayin S, Karabati S (1999) A Bicriteria Approach to the Two-Machine Flow Shop Scheduling Problem. Eur J Oper Res 113:435–449
84. Shyu SJ, Lin BMT, Yin PY (2004) Application Of Ant Colony Optimization For No-Wait Flowshop Scheduling Problem To Minimize The Total Completion Time. Comput Indust Eng 47(2–3):181–193
85. Smutnicki C (1998) Some Results of the Worst-Case Analysis for Flow Shop Scheduling. Eur J Oper Res 109:66–87
86. Solimanpur M, Vrat P, Shankar R (2004) A Neuro-Tabu Search Heuristic for the Flow Shop Scheduling Problem. Comput Oper Res 31:2151–2164
87. Soukhal A, Oulamara A, Martineau P (2005) Complexity of Flow Shop Scheduling Problems with Transportation Constraints. Eur J Oper Res 161:32–41
88. Steinhöfel K, Albrecht A, Wong CK (2002) The Convergence of Stochastic Algorithms Solving Flow Shop Scheduling. Theoretic Comput Sci 285:101–117
89. Suliman SMA (2000) A Two-Phase Heuristic Approach to the Permutation Flow-Shop Scheduling Problem. Int J Prod Econ 64:143–152
90. Suresh V (1997) A Note on Scheduling of Two-Stage Flow Shop with Multiple Processors. Int J Prod Econ 49:77–82
91. Szwarc W (1983) Flowshop problems with time lags. Manag Sci 29:477–481
92. Tang L, Xuan H, Liu J (2006) A New Lagrangian Relaxation Algorithm for Hybrid Flowshop Scheduling to Minimize Total Weighted Completion Time. Comput Oper Res 33(11):3344–3359
93. Tasgetiren MF, Liang YC, Sevkli M, Gencyilmaz G (2007) A Particle Swarm Optimization Algorithm for Makespan and Total Flowtime Minimization in the Permutation Flowshop Sequencing Problem. Eur J Oper Res 177(3):1930–1947
94. Tian P, Ma J, Zhang D-M (1999) Application of the Simulated Annealing Algorithm to the Combinatorial Optimisation Problem with Permutation Property: An Investigation of Generation Mechanism. Eur J Oper Res 118:81–94
95. Toktaç B, Azizoğlu M, Köoksalan SK (2004) Two-Machine Flow Shop Scheduling with Two Criteria: Maximum Earliness and Makespan. Eur J Oper Res 157:286–295
96. Townsend W (1977) Sequencing n-Jobs on m-Machines to Minimize Maximum Tardiness: A Branch-and-Bound Solution. Manag Sci 23:1016–1019
97. Vallada E, Ruiz R, Minella G (2008) Minimising Total Tardiness in the M-Machine Flowshop Problem: A Review

and Evaluation of Heuristics and Metaheuristics. Comput Oper Res 35(4):1350–1373

98. Varadharajan TK, Rajendran C (2005) A Multi-Objective Simulated-Annealing Algorithm for Scheduling in Flowshops to Minimize the Makespan and Total Flowtime of Jobs. Eur J Oper Res 167(3):772–795
99. Wang C, Chu C, Proth J-M (1996) Efficient Heuristic and Optimal Approaches for n/2/F/$\sum C_i$ Scheduling Problems. Int J Prod Econ 44:225–237
100. Wang C, Chu C, Proth J-M (1997) Heuristic Approaches for n/m/F/$\sum C_i$ Scheduling Problems. Eur J Oper Res 96:636–644
101. Wang J-B (2007) Flow Shop Scheduling Problems with Decreasing Linear Deterioration Under Dominant Machines. Comput Oper Res 34(7):2043–2058
102. Wang J-B, Xia Z-Q (2006) Flow Shop Scheduling with Deteriorating Jobs under Dominating Machines. Omega 34(4):327–336
103. Wang L, Zhang L (2006) Stochastic Optimization Using Simulated Annealing with Hypothesis Test. Appl Math Comput 174(2):1329–1342
104. Wardono B, Fathi Y (2004) A Tabu Search Algorithm for the Multi-Stage Parallel Machine Problem with Limited Buffer Capacities. Eur J Oper Res 155(2):380–401
105. Xuan H, Tang L (2007) Scheduling a Hybrid Flowshop with Batch Production at the Last Stage. Comput Oper Res 34(9):2718–2733
106. Yokoyama M (2001) Hybrid Flow-Shop Scheduling with Assembly Operations. Int J Prod Econ 73:103–116
107. Yokoyama M, Santos DL (2005) Three-Stage Flow-Shop Scheduling with Assembly Operations to Minimize the Weighted Sum of Product Completion Times. Eur J Oper Res 161:754–770

Forecasting

Ulrike Leopold-Wildburger[1], Otwin Becker[2], Martin Kuehrer[3]

[1] Department of Statistics and OR,
Karl-Franzens University of Graz, Graz, Austria

[2] Institute for Alfred Weber for Economics and Social Sciences, Faculty of Economics and Social Studies, University of Heidelberg, Heidelberg, Germany

[3] Siemens AG (NYSE: SI), Wien, Austria

MSC2000: 90C30, 90C26

Article Outline

Keywords
Introduction
Forecasting Methods and Models
  Expert Systems
  Subjective Curve Fitting
  Technological Comparison
  Expectations and Decision
  Statistical Procedures
  Moving Average Model
  Regression Analysis
  Econometric Models
Modern Computer Aided Techniques
  Neural Networks
  Fuzzy Logic
  Genetic Algorithms
See also
References

Keywords

Artificial intelligence; Curve fitting; Econometrics; Expert systems; Exponential smoothing; Extrapolation; Fuzzy logic; Genetic algorithms; Inductive inference; Moving average model; Neural networks; Regression analysis; Statistics; Time series; Time series analysis; Visual inference

This article consists of three parts: first, a historical introduction to the topic, next an overview of the most frequent forecasting methods and finally a short description of modern computer-aided techniques as they are used nowadays (2000) for instance for forecasts on financial markets.

Introduction

Prediction ideas and information about uncertain future events in general are as old as humanity. Scientific forecasts are based on predetermined patterns, regularities or conformities with a (natural) law. A theoretical basis is made up of its components, the parameters of the model and the conditions for the system.

Predictions of future events are called forecasts and are concerned with the question of what the world 'will' look like [6]. Any organization must be able to make forecasts concerning their work which aim to reduce the uncertainty of the environment. For example, business firms, in particular, require forecasts for a large number of events and conditions in all phases of their operation and forecasts are indispensable for planning and strategy in everyday life.

For centuries, the nature of forecasting was the field of philosophers, who studied problems of *inductive inference* analytically in order to obtain instruments for qualitative and quantitative forecasting.

When asking the question of whether the future can be predicted, or whether it is arbitrary and random, we first noticed certain regularities in the behavior of nature. These regularities were most obvious in the fields of physics and astronomy in which it was possible to make conditional forecasts. They were given their firm mathematical basis by I. Newton, in the seventeenth century. We still use his theory of gravity which is able to predict the motion of (celestial) bodies.

At the turn into the twentieth century, psychology began to use experimental methods to investigate learning in humans and other organisms. In doing so, psychologists acquired knowledge about which behavior could be forecasted and how to reduce uncertainty as much as possible. During the twentieth century, the topic of forecasting in general became increasingly important, especially after quantitative methods had been developed. Various forecasting methods were given priority in economics and even more recently the computer has provided research tools, engendering the field of machine learning. Systematic research on trade cycles and on crises management are the first economic forecasts, at the same time as early psychological investigations.

One of the first aims of economics was to become a science which could make forecasts with the help of induction. The true measure of the value of economists is often seen as the accuracy of the forecasts they make [14].

J.H. Holland, K. Holyoak, R. Nisbett and P. Thagard in 1986 [18] gave an excellent overview of the various insights of researchers in psychology, philosophy and *artificial intelligence*. Also borrowing from several other disciplines such as engineering, *statistics*, biology and game theory, including experimental economics [20] they systematically developed principles providing coherence of a diverse set of findings on the nature of inductive processes for prospective events in the future.

Forecasting Methods and Models

Obviously there are several possibilities of classification because of various methodological approaches. Using a fundamental division, we will generate two groups of forecasting methods:

i) qualitative methods and
ii) quantitative methods.

Another main distinction consists of a generalization of similar situations which can be

i) data based (usually given in the form of a time series, a chronological sequence of observed data with respect to a certain variable) expecting that history repeats itself in a certain way and
ii) theory based, where we assume that external factors determine events.

It is natural to start with *qualitative forecasting methods* predicting future events with a certain subjective probability: on the one hand we tend to make forecasts for similar events on the basis of a certain generalization, on the other hand we try to predict new events for those situations where little or no historical data are available and for events where we expect changes within the data patterns. Generalization ideas - in a logical and methodological sense - are made on the basis that events will have properties in a certain analogy to the past and tend towards the direction of objective processes.

Here we want to recommend a well known classification by S. Makridakis and S.C. Wheelwright [21]. Our aim is to discuss a selected subset of these frequently used methods.

Expert Systems

In questions about future events, a systematic discussion of a group of five to twelve experts (*expert systems*) usually yields forecasts with a better hit-rate than individual predictions. This belongs to the class of *judgemental forecast*. Using this method, credibility is one of the most desirable features of a forecast [10].

The *Delphi method*, developed by members of the RAND Corporation in the 1960s [11], avoids face-to-face effects by using a procedure based on a questionnaire technique. Delphi therefore guarantees three basic characteristics: anonymity, interaction with controlled feedback, and statistical group responses.

Subjective Curve Fitting

A frequently used method is *subjective curve fitting and extrapolation*, which is used in economics, for example, for forecasts on the development of products with certain life cycles or seasonal fluctuations. Experimental findings show that there are several gestaltsoriented rules dominating expectation formations [4,5]. Subjective curve fitting differs to some extent between the different subjects; frequently we are not only interested in individual expectations of a single forecasting subject but also interested in the so-called average opinion of a whole group as a good predictor for future events. For example it is a well known hypothesis that the average expected rate of inflation has an essential influence on various economic variables.

Technological Comparison

A sensible method is *technological comparison* [2] and [15]. Obviously, we should enlarge, compare or combine these qualitative methods with quantitative ones (combined forecasts), knowing that even models which best fit the available data are not necessarily the most accurate ones in predicting beyond this data [23].

Expectations and Decision

The simplest way of modeling expectations of future events, which is used frequently in economic theory, is to assume that conditions prevailing today will be maintained in all subsequent periods analogously. In cases where no causal explanation from other variables seems to be appropriate we could simply use *extrapolation methods* with the given data base to enable at least a short term forecast. These methods are successfully used

i) for seasonally adjusted data; and
ii) for cases where a continuation of the historical trend is to be expected.

Statistical Procedures

The next step is to use statistical procedures which are in some sense learning and in another sense adaptive methods. *Quantitative forecasting methods*, theory and data based, using knowledge from mathematical statistics started to be successful in the early 1960s beginning with ideas of R.G. Brown on *smoothing methods*.

In particular, *exponential smoothing* is frequently used for producing short term forecasts [8,9]. Brown suggested estimating the average of a time series and used it as an *extrapolation* for the forecast. With each new data set and observation respectively, we are able to revise the mean square error (MSE) applying exponential smoothing to the squares of the error in the most recent forecast. Several techniques have been proposed using *exponential smoothing* but it is evident that all the history of a process cannot be described by one and the same simple model.

Moving Average Model

In 1970 G.E. Box and G.M. Jenkins introduced more sophisticated forecasting models which were the first to take into account the nature of the data and the manner of the stochastic process to be forecasted. They asked not only the question of what to forecast and what data to collect, but also what data to analyze and in what context to embed the forecast. Their *moving average models* [7], enable a successful application. They popularized an approach that combines the moving average and the autoregressive approaches in [7]. The classes of autoregressive (integrated) moving averages (ARMA and ARIMA) processes have been successfully introduced by them and their models are some of the most frequently used tools for stochastic analysis.

Several ways to model multiple time series are described in [16]. Further ARIMA models are given in [23,24]; [13] gives an excellent comparison of these models using performance methods.

When enlarging the statistical methods with sensible associations and connections, *econometric methods* should be considered, if causal relationship and changes in causal variables are expected and can be estimated.

Regression Analysis

Usually it would be sensible to figure out a certain a priori relationship between the given data sets such that statistical methods of regression analysis can be used. In its simplest form, the *classical linear regression model* is used to determine an equation relating two sets of data with each other:

i) the set of observations of the explanatory or independent variables (the predictors); and
ii) the set of the associated responses, the observations of the dependent variables.

This task often seems to be easy at first sight, but when all details are concerned it becomes a high leveled task.

Obviously there are some future values which can easily be forecasted, e. g. the fuel consumption for a certain period as a relation of velocity.

As our example from the field of finance will demonstrate, there are, however, enough reasons to assume more complicated situations caused by complex systems and/or error terms. For example, demand as the variable of interest can be seen as a function of the price which takes the role of the predictor.

Econometric Models

Obviously, the standard model using the single regression equation has been extended in various ways. For example,

i) disturbances are allowed to be autocorrelated and to have different variances (heteroskedastic);
ii) by regressors measured with errors or in dynamic situations (with dependences on lagged values) stochastic regressors arise.

As proposed already at the foundation of the Econometric Society in Cleveland in 1930 we also use nowadays (2000) econometric models which are not only data based, but also theory oriented.

In econometrics, the single equation regression model is enlarged and complex systems of *simultaneous equation models* are used, including several equations and also several dependent variables. These models are implemented as applied econometrics software and build, for instance, the basis for national budget calculations, usually containing several hundred equations.

Modern Computer Aided Techniques

To predict future movements of financial markets, technical analysts use time series and apply the statistical and econometrical methods described above. We enlarge the methods by new techniques which are able to recognize certain relationships from examples by generalization with the help of new computer technologies. The methods used in this application are a composition of artificial neural networks, genetic algorithms (see ► Genetic algorithms) and fuzzy logic. Obviously we are not able to go into details, but we try to give a short characterization for our application.

Neural Networks

These are inspired by the functionality of nerve cells in the brain. Like humans, they can learn to recognize patterns by repeated exposure to many different examples. They can be used to detect salient characteristics whether they are handwritten characters, profitable loans or good trading decisions. Neural networks learn to recognize even regularities in data that are inexact or incomplete. A neural network finds this relationship by means of a learning cycle where a large amount of samples are presented repeatedly to the network. Neural networks cannot guarantee an optimal solution to a problem. However, properly configured and trained neural networks can often make consistently good classifications, generalizations or decisions in a statistical sense. Neural networks are widely used to identify patterns in the data of financial markets.

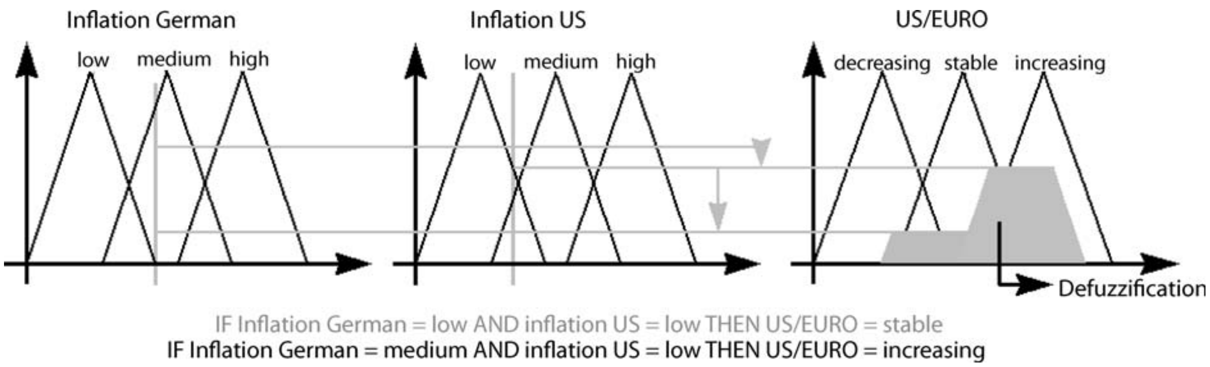
Fuzzy Logic

This is a strategy that is not based on a mathematical description of a special system or market but is intended to model the behavior of a human investor. The expert's knowledge is specified in terms of linguistic rules in which linguistic expressions are associated with fuzzy sets. Fuzzy set methods tend to overcome the vagueness of causality. They can be used to explain financial markets' developments using fundamental rules as shown in Fig. 1.

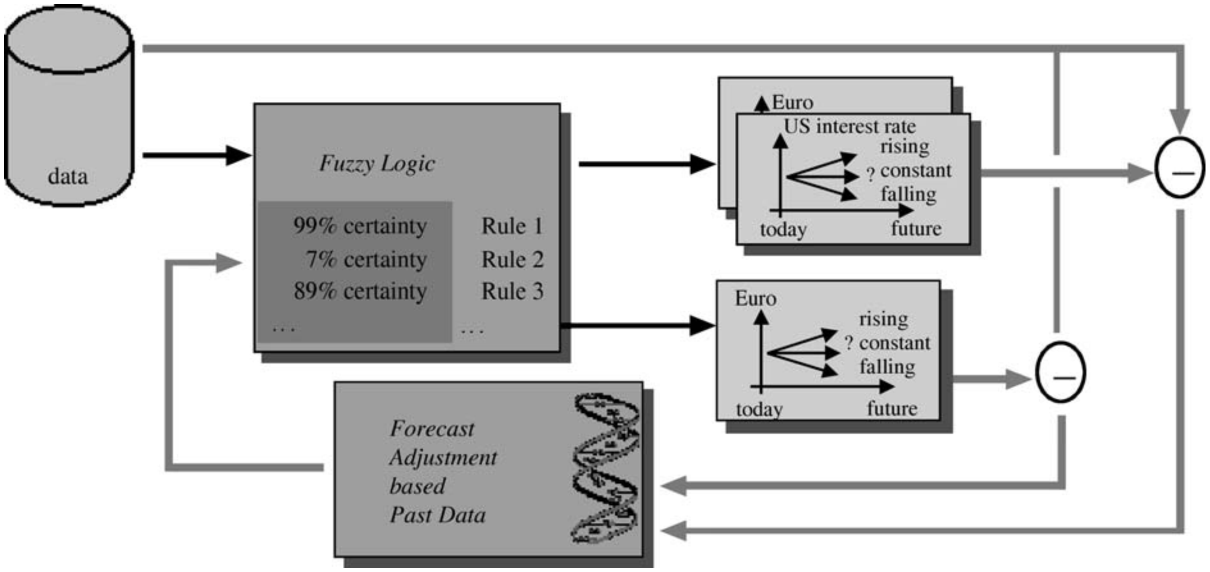
Fuzzy logic is a superset of conventional (Boolean) logic that has been extended to handle the concept of partial truth – truth values between 'completely true' and 'completely false'. [25,26]. In other words, a fuzzy system is a collection of 'membership functions' and rules that are used to reason about data. In our example the ranges of the interest rates of Germany and the USA gives forecasts for the exchange rate between these currencies. Fuzzy logic enables us to model and predict market developments on the basis of the experience of financial experts.

Genetic Algorithms

A genetic algorithm allows us to optimize any given function. Genetic algorithms are search procedures based on the mechanisms of natural selection, mutation


Inflation German
low
medium
high
Inflation US
low
medium
high
US/EURO
decreasing
stable
increasing
Defuzzification
IF Inflation German = low AND inflation US = low THEN US/EURO = stable
IF Inflation German = medium AND inflation US = low THEN US/EURO = increasing

Forecasting, Figure 1
Fuzzy Logic rules to predict the US Dollar/Euro fixingj



Forecasting, Figure 2
Genetic algorithm used for optimizing a rulebasis to forecast the US Dollar/Euro fixing

and recombination. A population consisting of chromosomes (i. e., solutions for a function) is created randomly. In the next step each chromosome is evaluated and given a certain fitness value. The fitness value represents the feasibility and the optimality of a given solution. Depending on their fitness value a certain percentage of the population is selected and deleted. The surviving individuals are recombined and mutated. After the population has been evaluated and the forecast adjustment based on past data has been decided, the selection process starts again.

The genetic algorithm is used to optimize the certainty of each rule in the *fuzzy logic* rulebase. As shown in Fig. 2 the fitness function of the genetic algorithm consists of a fuzzy logic rulebase and several mathematical objects to calculate the forecast error. The forecast error is used to evaluate the individuals. All rules are applied to historic financial data and their forecast error is summed up over the whole horizon.

Rules which are only partly true get lower certainty values until their certainty corresponds to their actual influence. Currency markets tend to follow certain regularities, detected by expert knowledge or, for example, the purchasing power theorem.

A similar procedure is applied to neural networks in order to optimize the topology of the neural network itself as well as a data mining approach to identify the input parameters. The genetic algorithm allows us to cancel out useless time series. When forecasting financial markets an appropriate and adaptive input parameter selection is necessary.

In our case the inputs are knowledge of economic data to receive forecasts for future developments of financial markets.

One goal in system theory is, in order to integrate the ideas of several disciplines, to have a successful instrument for analyzing forecasting processes including learning and discovery in direction optimality. This process of searching for the best value that can be realized or attained is based on the events of subjects whose actions are not able to be forecasted with total certainty.

Finally, we are able to summarize this as follows, using the different stages we took into account:

- historical comparison based on repeatedly observed similar events and on statistical data, e. g. business fluctuations, Harvard's barometers, chart extrapolations;
- time series analysis, based on proceedings on mathematical statistics;
- econometric forecasting models, including stepwise regression models, as well as vector autoregressive models (VAR);
- modern techniques, mainly computer aided, and software which is available as adaptive models, learning models, artificial neural networks (ANN), fuzzy set models and evolutionary algorithms.

See also

► Continuous Global Optimization: Applications

References

1. Armstrong JS (1985) Long range forecasting: From crystal ball to computer, 2nd edn. Wiley, New York
2. Armstrong JS (1996) Forecasting. In: Gass S, Harris C (eds) Encycl. Oper. Res. and Management Sci. Kluwer, Dordrecht, pp 229–233
3. Armstrong JS, Collopy F (1992) Error measures for generalizing about forecasting methods: Empirical comparison. Internat J Forecasting 8:69–80
4. Becker O, Leopold-Wildburger U (1996) The bounds & likelihood-procedure - A simulation study concerning the efficiency of visual forecasting techniques. Central Europ J Oper Res and Economics 4(2/3):223–229
5. Becker O, Leopold-Wildburger U (2000) Erwartungsbildung und Prognose. Austrian J Statist:29(1):1–76
6. Bowerman B, O'Connell R (1993) Forecasting and time series: An applied approach, 3rd edn. Duxbury, Belmont, CA
7. Box GE, Jenkins GM (1970) Time series analysis for forecasting and control. Holden Day, San Francisco
8. Brown RG (1959) Statistical forecasting for inventory control. McGraw-Hill, New York
9. Brown RG (1962) Smoothing, forecasting and prediction of discrete time series. Prentice-Hall, Englewood Cliffs, NJ
10. Bunn DW (1996) Non-traditional methods of forecasting. Europ J Oper Res 92:528–536
11. Dalkey N, Helmer O (1963) An experimental application of the Delphi method to the use of experts. Managem Sci 9:458–467
12. Davis L (ed) (1991) Handbook of genetic algorithms. v. Nostrand, Princeton, NJ
13. Fildes R, Hibon M, Makridakis S, Meade N Generalizing about univariate forecasting models, WP Imperial College,
14. Friedman M (1963) Individual behaviour in oligopolistic markets: An experimental study. Yale Economic Essays 3:359–417
15. Gerstenfeld A (1944) Technological forecasting. J Business 44:10–18
16. Granger GWC, Newbold P (1977) Forecasting economic time series. Acad. Press, New York
17. Holland J (1975) Adaptation in natural and artificial systems. Univ. Michigan Press, Ann Arbor, MI
18. Holland J, Holyoak K, Nisbett R, Thagard P (1986) Induction, processes of inference, learning and discovery. MIT, Cambridge, MA
19. Kohonen T (1997) Self organizing maps, 2nd edn. Springer, Berlin
20. Leopold-Wildburger U (1997) Some selected topics in experimental economics. Internat Trans Oper Res 4(3):165–174
21. Makridakis S, Wheelwright SC (1987) The handbook of forecasting: A manager's guide, 2nd edn. Wiley, New York
22. Makridakis S, Wheelwright SC (1989) Forecasting methods for management, 3rd edn. Wiley, New York
23. Makridakis S, Winkler RL (1983) Averages of forecasts: Some empirical results. Managem Sci 29:987–996
24. Montgomery DC, Johnson LA, Gardiner JS (1990) Forecasting and time series analysis, 2nd edn. McGraw-Hill, New York
25. Yager RR, Zadeh LA (1995) An introduction to fuzzy logic applications in intelligent systems, 3rd edn. Kluwer, Dordrecht
26. Zimmermann H-J (1996) Fuzzy set theory, 3rd edn. Kluwer, Dordrecht

Fourier–Motzkin Elimination Method

LEONID KHACHIYAN
Department Computer Sci., Rutgers University,
Piscataway, USA

MSC2000: 52B12, 68Q25

Article Outline

Keywords
Solution of Systems of Linear Inequalities
and Linear Programming Problems
Complexity of the Fourier–Motzkin Method
See also
References

Keywords

Fourier–Motzkin elimination; Linear inequalities; Linear programming; First order theory of real addition with order; Semilinear set; Projection; Output-polynomial

Consider a system of m linear inequalities in n real variables

$$Ax \le b, \tag{1}$$

where $x = (x_1, \ldots, x_n)^\intercal \in \mathbf{R}^n$ is the vector of unknowns and A, b are a given real matrix and vector. Let $X = \{x \in \mathbf{R}^n : Ax \le b\}$ be the solution set of the system, and let $X^{[k]}$ denote the projection of X onto the linear space spanned by the last $n - k$ coordinates:

$$\begin{aligned} &X^{[k]} = \{(x_{k+1}, \ldots, x_n) \in \mathbb{R}^{n-k} : \\ &\exists (x_1, \ldots, x_k) \in \mathbb{R}^k \\ &\text{s.t. } (x_1, \ldots, x_n) \in X\}. \end{aligned}$$

The *Fourier–Motzkin method* [3,4,5,8,10,12,14,15] successively eliminates variables $x_1, \ldots, x_{n-1}$ from (1) and computes matrices $A^{[k]}$ and vectors $b^{[k]}$ such that

$$X^{[k]} = \left\{ x^{[k]} \in \mathbb{R}^{n-k} \colon \ A^{[k]} x^{[k]} \le b^{[k]} \right\}, \qquad k = 1, \ldots, n-1,$$

where $x^{[k]} = (x_{k+1}, \ldots, x_n)^\intercal$.

In order to eliminate variable x_1, we first multiply each of the m inequalities of (1) by an appropriate positive scalar to make each entry in the first column of A equal to $\pm$ 1 or 0. We can thus assume without loss of generality that the original system of inequalities has the form

$$\begin{aligned} +1 \cdot x_1 + \alpha_i(x^{[1]}) &\le 0, \quad i \in M_+, \\ -1 \cdot x_1 + \alpha_i(x^{[1]}) &\le 0, \quad i \in M_-, \\ 0 \cdot x_1 + \alpha_i(x^{[1]}) &\le 0, \quad i \in M_0, \end{aligned}$$

where $\alpha_i(x^{[1]}) = \alpha_{i2}x_2 + \cdots + \alpha_{in}x_n + \beta_i$ are given affine forms of $x^{[1]} = (x_2, \ldots, x_n)^\intercal \in \mathbf{R}^{n-1}$ and M_+, M_-, M_0 are disjoint sets of (indices of) inequalities partitioning the entire set of inequalities in (1):

$$M_+ \cup M_- \cup M_0 = \{1, \ldots, m\}.$$

It is easy to see that for each fixed $x^{[1]}$, the inequalities with indices $i \in M_+ \cup M_-$ can be satisfied by some real x_1 if and only if each upper bound $-\alpha_i(x^{[1])}$, $i \in M_+$ on x_1 exceeds each lower bound $\alpha_j(x^{[1]})$, $j \in M_-$ on the same variable, i. e., $-\alpha_i(x^{[1]}) \ge \alpha_j(x^{[1]})$ for all $i \in M_+$ and $j \in M_-$. Combining these $|M_+|\,|M_-|$ inequalities with the remaining $|M_0|$ inequalities of (1) that do not depend on x_1, we arrive at the system of $|M_+|\,|M_-| + |M_0|$ linear inequalities

$$\begin{aligned} &\alpha_i(x^{[1]}) + \alpha_j(x^{[1]}) \le 0, \quad (i, j) \in M_+ \times M_-, \\ &\alpha_i(x^{[1]}) \le 0, \quad i \in M_0, \end{aligned}$$

whose solutions set is $X^{[1]}$. The above system can be written as $A^{[1]}x^{[1]} \le b^{[1]}$ with appropriate matrix $A^{[1]}$ and vector $b^{[1]}$. This gives $X^{[1]} = \{x^{[1]} \in \mathbf{R}^{n-1} : A^{[1]}x^{[1]} \le b^{[1]}\}$. Eliminating variable x_2 from $A^{[1]}x^{[1]} \le b^{[1]}$ we obtain a similar description $X^{[2]} = \{x^{[2]} \in \mathbf{R}^{n-2} : A^{[2]}x^{[2]} \le b^{[2]}\}$ for the second projection and so on. After $n - 1$ steps of the above procedure we have $n - 1$ matrices $A^{[k]}$ and vectors $b^{[k]}$ such that $X^{[k]} = \{x^{[k]} \in \mathbf{R}^{n-k} : A^{[k]}x^{[k]} \le b^{[k]}\}$, $k = 1, \ldots, n - 1$.

Solution of Systems of Linear Inequalities and Linear Programming Problems

If the solution set $X = \{x \in \mathbf{R}^n \colon Ax \le b\}$ is nonempty, then so are all the projections $X^{[k]} \subseteq \mathbf{R}^{n-k}$, $k = 1, \ldots, n - 1$, and vice versa. In particular, if $Ax \le b$ is feasible, then

$$X^{[n-1]} = \left\{ x^{[n-1]} \in \mathbb{R} \colon \ A^{[n-1]} x^{[n-1]} \le b^{[n-1]} \right\}$$

is a nonempty interval on the scalar variable $x^{[n-1]} = x_n$. Given $A^{[n-1]}$ and $b^{[n-1]}$, we can easily find a point $\overline{x}_n \in X^{[n-1]}$. Then, substituting $x_n = \overline{x}_n$ into $A^{[n-2]}x^{[n-2]} \leq b^{[n-2]}$, we obtain a new feasible system of linear inequalities whose solution set is the interval $\left\{x_{n-2} \in \mathbb{R}\colon\ (x_{n-1}, \overline{x}_n) \in X^{[n-2]}\right\}$. Solving this one-variable system yields a point $\overline{x}^{[n-2]} = (\overline{x}_{n-1}, \overline{x}_n) \in X^{[n-2]}$, which can be substituted in $A^{[n-3]}x^{[x-3]} \leq b^{[n-3]}$ etc. By repeating such backward substitutions, the Fourier–Motzkin method can compute a solution $(\overline{x}_1, \dots, \overline{x}_n)$ to any feasible system of linear inequalities $Ax \leq b$. 'Historically, it is the 'pre-linear programming' method to solve linear inequalities' [14].

Now suppose that the input system is infeasible, i. e. $X = \{x \in \mathbf{R}^n\colon Ax \leq b = \emptyset\}$. As was pointed out in [10], the Fourier–Motzkin method can then find nonnegative real multipliers $p_1, \dots, p_m$ such that

$$pA = 0, \quad pb = -1, \quad p = (p_1, \dots, p_m) \geq 0. \quad (2)$$

To see this, observe that each inequality in $A^{[1]}x^{[1]} \leq b^{[1]}$ is a positive combination of at most two inequalities of the original system. Since a nonnegative combination of nonnegative combinations of some inequalities is a nonnegative combination of the same inequalities, we conclude that each inequality in each system $A^{[k]}x^{[k]} \leq b^{[k]}$, $k = 1, \dots, n-1$, is a nonnegative combination of the input inequalities. Considering that $A^{[n-1]}x^{[n-1]} \leq b^{[n-1]}$ is an infeasible system of linear inequalities in one variable, $A^{[n-1]}x^{[n-1]} \leq b^{[n-1]}$ is easily seen to contain one or two inequalities whose positive combination yields the infeasible inequality $0 \cdot x_n \leq -1$. This is equivalent to (2). In particular, the Fourier–Motzkin method provides a simple algorithmic proof of the Farkas lemma (cf. ► Farkas lemma; ► Farkas lemma: Generalizations): (1) is feasible if and only if (2) is infeasible.

The Fourier–Motzkin method can also be used to solve the general linear programming problem

$$\xi^* = \max\left\{c^\top x\colon\ Ax \leq b,\ x \in \mathbb{R}^n\right\}. \quad (3)$$

For instance, we can eliminate n variables $x = (x_1, \dots, x_n)$ from $Ax \leq b$, $x_{n+1} - c^\top x \leq 0$ to determine the interval $X^{[n]} = \{x_{n+1} : x_{n+1} \leq \xi^*\}$. Then, letting $x_{n+1} = \xi^*$ and solving the resulting system yields an optimal solution.

It should be mentioned that there are far more efficient linear programming algorithms. Note, however, that (3) calls for projecting $X = \{x \in \mathbf{R}^n\colon Ax \leq b\}$ on a one-dimensional subspace. After an appropriate linear transformation, the Fourier–Motzkin method can project $X = \{x \in \mathbf{R}^n\colon Ax \leq b\}$ on any given subspace in $\mathbf{R}^n$.

Complexity of the Fourier–Motzkin Method

Let m_k denote the number of inequalities in the kth system $A^{[k]}x^{[k]} \leq b^{[k]}$ generated by the Fourier–Motzkin method. Since $m_1 = |M_+|\ |M_-| + |M_0| \leq m^2$, we have $m_k \leq m_{k-1}^2$ for all k. So the number of inequalities is at most squared at each step of the method, which implies that m_k is bounded by a doubly exponential function in k, say $m_k \leq m^{2^k}$. The following example shows that with sufficiently many variables, the kth step of the method can produce

$$m_k = m^{2^{k(1-o(1))}}$$

inequalities.

Example 1 [14] Let $n = 2^k + k + 2$ and consider a system of linear inequalities $Ax \leq b$ which contains as left-hand sides $m = 8\binom{n}{3}$ linear forms $\pm x_{i_1} \pm x_{i_2} \pm x_{i_3}$ for all $1 \leq i_1 < i_2 < i_3 \leq n$. By induction on $j = 1, \dots, k$ it is easy to show that after eliminating the first j variables, the resulting system includes among its left-hand sides all the forms $\pm x_{i_1} \pm \cdots \pm x_{i_s}$ with $k + 1 \leq i_1 < \cdots < i_s \leq n$ and $s = 2^j + 2$. In particular, for $j = k$ we have at least $2^{2^k+2} = m^{2^{k(1-o(1))}}$ inequalities in $A^{[k]}x^{[k]} \leq b^{[k]}$.

Let us now return to the first step of the algorithm where we replace $Ax \leq b$ by the $|M_+|\ |M_-|+|M_0|$ new inequalities $A^{[1]}x^{[1]} \leq b^{[1]}$. As was pointed out already by J.B.J. Fourier, 'it nearly always happens that a rather large number of these new inequalities are redundant' and 'their removal greatly simplifies the problem' [8]. If the redundant inequalities are systematically removed at each step of the algorithm, the number m_k of inequalities generated by kth step of the Fourier–Motzkin method is bounded by an exponential function in k. Assume without loss of generality that $X = \{x \in \mathbf{R}^n\colon Ax \leq b\}$ is full-dimensional, then each projection $X^{[k]}$ is also full-dimensional and m_k is the number of facets of $X^{[k]}$. Therefore m_k is bounded by the total number of i-faces of X for $i \geq n - k - 1$. Hence

$$m_k \leq \sum_{i=1}^{k+1} \binom{m}{i} \sim \frac{m^{k+1}}{(k+1)!} \quad \text{for } m \to \infty.$$

(This rough estimate can be improved by using the upper bound theorem [11]; in particular, m_k cannot grow faster than $m^{\lfloor n/2 \rfloor}$.) In the example below, $X^{[k]}$ has

$$m_k \geq \frac{m^{k+1}}{(k+1)^{k+1}}$$

facets.

Example 2 Let $s \geq 2$ be a natural number. Consider the system of $m = (k+1)\, s$ linear inequalities

$$\begin{aligned} &y_{ij} \geq x_i, \quad i = 1,\dots,k, \quad j = 1,\dots,s,\\ &x_1 + \cdots + x_k \geq z_l, \quad l = 1,\dots,s, \end{aligned}$$

where x_i, y_{ij}, and z_l are real variables. The elimination of $x_1,\dots,x_k$ results in $s^{k+1} = (m/(k+1))^{k+1}$ inequalities

$$y_{1f(1)} + \cdots + y_{kf(k)} \geq z_l, \quad l = 1,\dots,s,$$

where f ranges over the set of all s^k mappings from $\{1, \dots, k\}$ to $\{1, \dots, s\}$. None of the inequalities above is redundant. For instance,

$$y_{11} + \cdots + y_{k1} \geq z_1$$

is violated by $y_{11} = \cdots = y_{k1} = 0$ and $z_1 = \cdots = z_s = 1$, whereas all the other inequalities can be satisfied by giving the remaining variables y_{ij} a high value.

Since detecting the redundancy of an inequality can be done via linear programming (or by maintaining a list of vertices and extreme directions of $X^{[k]}$ with the *double description method* [4,13], see also [9,15] and references herein), the Fourier–Motzkin method runs in exponential space and time. It is natural to ask whether given $X = \{x \in \mathbf{R}^n \colon Ax \leq b\}$ and a number $k \in \{1, \dots, n-1\}$, an irredundant description for $X^{[k]} = \{x^{[k]} \in \mathbf{R}^{n-\mathrm{k}}\colon A^{[k]}x^{[k]} \leq b^{[k]}\}$ can be computed in *output-polynomial time*, i. e. by an algorithm that runs in time polynomial in the total input and output size. This question is open even in the bit model of computation for rational A and b, when redundant inequalities can be detected in polynomial time. A related problem is the generation of all vertices for $X = \{x \in \mathbf{R}^n\colon Ax \leq b\}$. The vertex generation problem (or its dual, the *convex hull problem*) can also be solved by the double description method, see e. g. [1], but the question as to whether there is an output-polynomial vertex generation algorithm remains open.

Finally, we mention that the Fourier–Motzkin method can be modified to a quantifier-elimination method for arbitrary *semilinear sets*

$$\begin{aligned} \mathcal{X}^{[k]} = \{(x_{k+1},\dots,x_n) \in \mathbb{R}^{n-k}\colon &\\ (Q_1x_1 \in \mathbb{R})\cdots(Q_kx_k \in \mathbb{R})&\\ \mathcal{F}(x_1,\dots,x_n)\ \text{true}\},& \quad (4) \end{aligned}$$

where $Q_1, \dots, Q_k \in \{\exists, \forall\}$ are existential and/or universal quantifiers and $\mathcal{F}(x_1, \dots, x_n)$ is a given Boolean function of m threshold predicates

$$\mathcal{F}_i(x) = \begin{cases} \text{true} & \text{if } a_i^\top x \leq b_i,\\ \text{false} & \text{otherwise,} \end{cases}$$

with given coefficients $a_i \in \mathbf{R}^n$ and $b_i \in \mathbf{R}$, $i = 1, \dots, m$. In particular, if $Q_1, \dots, Q_k$ are all existential quantifies and $\mathcal{F} = \mathcal{F}_1 \wedge \cdots \wedge \mathcal{F}_m$, we obtain the previously considered problem of projecting the polyhedral set $X = \{x \in \mathbf{R}^n\colon a_i^\top x \leq b_i,\ i = 1, \dots, m\}$ onto the space spanned by the last $n - k$ coordinates. In general, (4) can be transformed into an equivalent quantifier-free representation $\mathcal{X}^{[k]} = \{(x_{k+1},\dots, x_n) : \mathcal{G}(x_{k+1},\dots, x_n)\ \text{true}\}$, where $\mathcal{G}$ is some Boolean formula whose atoms are new threshold predicates of $(x_{k+1},\dots, x_n) \in \mathbf{R}^{n-k}$. This can be done, for instance, as follows [6,7]. To eliminate the rightmost quantifier $Q_kx_k \in \mathbf{R}$, write each threshold inequality involving x_k in the form $x_k \leq \alpha_i(x^{(k)})$ or $x_k \geq \alpha_i(x^{(\mathrm{k})})$, where the α_i′ are given affine forms of the remaining variables $x^{(\mathrm{k})} = (x_1, \dots, x_{k-1}, x_{k+1}, \dots, x_n)$. Replace the infinite range $x_k \in \mathbf{R}$ by the finite set S of sample points $x_k = (\alpha_i(x^{(k)}) + \alpha_j(x^{(\mathrm{k})}))/2$ and $x_k = \pm\infty$. Now it is easy to see that the expression $(\exists x_k \in \mathbf{R})\mathcal{F}(x_1, \dots, x_n)$ is equivalent to the quantifier-free disjunction $\vee_{x_k\in S}\, \mathcal{F}(x_1,\dots,x_n)$ and that $(\forall x_k \in \mathbf{R})\mathcal{F}(x_1,\dots,x_n)$ can be replaced by the equivalent conjunction $\wedge_{x_k\in S}\mathcal{F}(x_1, \dots, x_n)$. Quantifies $Q_{k-1}x_{k-1}, \dots, Q_1x_1$ can be eliminated in the same way. For a discussion of faster algorithms that eliminate *blocks* of consecutive identically quantified variables see [2].

See also

► Linear Programming

References

1. Avis D, Bremner B, Seidel R (1997) How good are convex hull algorithms. Comput Geom Th Appl 7:265–302

2. Basu S (1997) An improved algorithm for quantifier elimination over real closed fields. In: Proc. 38th IEEE Symp. Foundations of Computer Sci., pp 56–65
3. Chvátal V (1983) Linear programming. Freeman, New York
4. Dantzig GB, Eaves BC (1975) Fourier–Motzkin elimination and its dual. J Combin Th A 14:288–297
5. Dines LL (1918/9) Systems of linear inequalities. Ann of Math 2(20):191–199
6. Eaves BC, Rothblum UG (1992) Dines–Fourier–Motzkin quantifier-elimination and applications of corresponding transfer principles over ordered fields. Math Program 53:307–321
7. Ferrante J, Rackoff C (1975) A decision procedure for the first order theory of real addition with order. SIAM J Comput 1:69–76
8. Fourier JBJ (1973) Analyse de travaux de l'Académie Royale de Sci., pendant l'anné 1824. In: Kohler DA (ed) Oper Res, 10, pp 38–42, Transl. of a Report by Fouries on his work on linear inequalities.
9. Fukuda K, Prodon A (1996) Double description method revisited. Lecture Notes Computer Sci 1120:91–111
10. Kuhn HW (1956) Solvability and consistency for linear equations and inequalities. Amer Math Monthly 63:217–232
11. McMullen P (1970) The maximal number of faces of a convex polytope. Mathematika 17:179–184
12. Motzkin TS (1936) Beiträge zur Theorie der linearen Ungleichungen. Doktorarbeit, Univ. Basel. Transl. in Cantor D, Gordon B, Rothschild B (eds) (1983): Contribution to the theory of linear inequalities: Selected papers. Birkhäuser, pp 81–103
13. Motzkin TS, Raifa H, Thompson GL, Thrall RM (1953) The double description method. In: Kuhn HW, Tucker AW (eds) Contributions to the Theory of Games, vol II. Ann Math Stud. 28 Princeton Univ. Press, Princeton, pp 81–103
14. Schrijver A (1986) Theory of linear and integer programming. Wiley/Interscience, New York
15. Ziegler GM (1994) Lectures on polytopes. Springer, Berlin

Fractional Combinatorial Optimization

FCO

Tomasz Radzik
King's College London, London, UK

MSC2000: 90-08, 90C27, 90C32, 68Q25, 68R05

Article Outline

Keywords
The Binary Search Method (BSM)
The Newton Method (NM)
Megiddo's parametric search (MPS)
See also
References

Keywords

Fractional optimization; Cost-to-time ratio; Newton method; Dinkelbach method; Megiddo parametric search

A *fractional combinatorial optimization problem* (FCOP) is a combinatorial optimization problem with an objective function which is a ratio of two (nontrivial) functions. Instances of a FCOP can be expressed in the general form:

$$\begin{cases} \max & \frac{f(\mathbf{x})}{g(\mathbf{x})}, \\ \text{for} & \mathbf{x} \in \mathcal{X}, \end{cases} \tag{1}$$

where $\mathcal{X} \subseteq \{0, 1\}^p$ is a set of (vectors representing) certain combinatorial structures, and f and g are real-valued functions defined on $\mathcal{X}$. Numbers $f(\mathbf{x})$, $g(\mathbf{x})$, and $f(\mathbf{x})/g(\mathbf{x})$ are usually called the *cost*, the *weight*, and the *mean-weight cost* of structure $\mathbf{x}$. A minimization FCOP is equivalent to the corresponding maximization problem, if the cost function f can be replaced with function $-f$. The FCOPs which appear in the literature on combinatorial optimization include: the *minimum ratio spanning-tree* problem [2,13,14]; the *maximum profit-to-time ratio cycle* problem and the equivalent *minimum cost-to-time ratio cycle* problem [1,3,6,11,12,13,14]; the *minimum mean cycle* problem [1,10,11]; the *maximum mean-weight cut* problem [16]; the *maximum mean cut* problem [5,9]; and the *fractional 0–1 knapsack* problem [7,8].

Consider, as an example, the minimum cost-to-time ratio cycle problem (MRCP). An instance of this problem consists of a directed graph $G = (V, E)$, where $E = \{e_1, \dots, e_m\}$ is the set of edges, and numbers c_i and t_i associated with each edge e_i, for $i = 1, \dots, m$. The objective is to find a simple cycle Γ in G which minimizes the ratio of $\sum\{c_i : e_i \in \Gamma\}$ to $\sum\{t_i : e_i \in \Gamma\}$. To express this instance of the MRCP in the form (1), let $\mathcal{X} \subseteq \{0, 1\}^m$ be the set of the characteristic vectors of the simple cycles in G, and for $\mathbf{x} = (x_1, \dots, x_m) \in \{0, 1\}^m$, let $f(\mathbf{x}) = -(c_1x_1 + \cdots + c_mx_m)$ and $g(\mathbf{x}) = t_1x_1 + \cdots + t_mx_m$. The MRCP models the following *tramp steamer* *problem* [1,12]: V is the set of ports which can be visited by our cargo ship; $E \subseteq V \times V$ is the set of possible direct port-to-port trips; numbers c_i and t_i are the cost and the transit time of trip $e_i \in E$, respectively; and the objective is to find a closed tour for the ship which minimizes the daily cost (or, equivalently, maximizes the daily profit).

A FCOP such that the denominator of the objective function $g(x_1, \dots, x_p) = x_1 + \cdots + x_p$ is commonly called a *uniform fractional combinatorial optimization* problem. The minimum mean cycle problem (which is a special case of the MRCP with all numbers t_i equal to 1) is a uniform FCOP. A FCOP such that $f(x_1, \dots, x_p) = a_1x_1 + \cdots + a_px_p$ and $g(x_1, \dots, x_p) = b_1x_1 + \cdots + b_px_p$ is called a *linear fractional combinatorial optimization* problem. All FCOPs mentioned above are linear.

Generic methods for FCO usually follow the *parametric approach to fractional optimization*. Let $\delta \in \mathbf{R}$ be a parameter. Problem:

$$\begin{cases} \max & f(\mathbf{x}) - \delta \cdot g(\mathbf{x}), \\ \text{for} & \mathbf{x} \in \mathcal{X}, \end{cases} \tag{2}$$

is called the *parametric problem* corresponding to the fractional problem (1). Let $h(\delta)$ denote the optimum objective value of problem (2). From now on assume that $f(\mathbf{x}) > 0$, for some $\mathbf{x} \in \mathcal{X}$, and $g(\mathbf{x}) > 0$, for all $\mathbf{x} \in \mathcal{X}$. Function h is continuous, convex, piecewise linear and strictly decreasing on $(-\infty, +\infty)$. It has exactly one root δ^* and this root is the optimum objective value of problem (1). The main generic methods for FCO are the *binary search method*, the *Newton method*, and *Megiddo's parametric search*. They all can be viewed as methods for finding the root of function h.

The Binary Search Method (BSM)

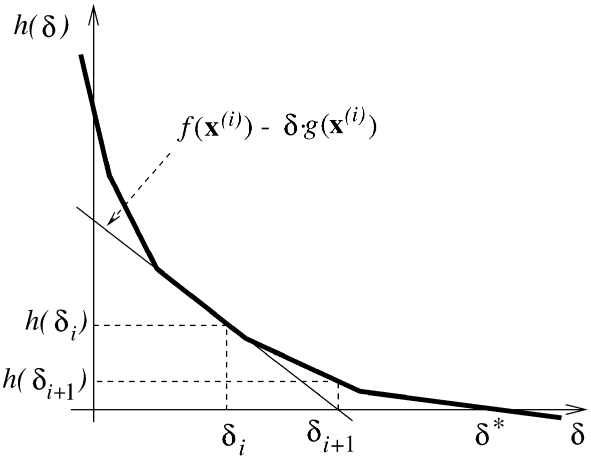
This method maintains an interval $[\alpha, \beta]$ containing the root δ^* of function h, and reduces this interval by half in each iteration by checking the sign of $h((\alpha+\beta)/2$. Thus to apply the BSM, one needs an algorithm $\mathcal{A}_0$ which for a given $\delta \in \mathbf{R}$ calculates the sign of the optimum objective value of problem (2). For a linear FCOP such that all numbers $|a_i|$ and $|b_i|$ are integers not greater than U (an *integral linear FCOP*), the BSM finds an optimum solution in $O(\log(pU))$ iterations. This follows form the fact that if numbers $f(\mathbf{x}')/g(\mathbf{x}')$ and $f(\mathbf{x}'')/g(\mathbf{x}'')$ are not equal, they must differ by at least $1/(pU)^2$. Hence, as soon as the length of the search interval $[\alpha, \beta]$ becomes less than $1/(pU)^2$, it contains only one value $f(\mathbf{x})/g(\mathbf{x})$, which must be equal to δ^*.

The Newton Method (NM)

This generic method for *fractional optimization*, also called the *Newton–Raphson method* or the *Dinkelbach method* [4], is an application of the classical Newton method to the problem of finding the root δ^* of function h. The NM computes an increasing sequence δ_1, $\delta_2, \dots$ of lower bounds on δ^*. During iteration i, a solution $\mathbf{x}^{(i)}$ of problem (2) for $\delta = \delta_i$ is computed, and δ_{i+1} is set to $f(\mathbf{x}^{(i)})/g(\mathbf{x}^{(i)})$ (see Fig. 1). The NM finds an optimum solution of a FCOP in a finite number of iterations, because function h consists of a finite number ($\leq |\mathcal{X}|$) of linear segments. The NM solves a uniform FCOP in at most $p + 1$ iterations, because function h corresponding to such a problem consists of at most p linear segments (since function g yields at most p different values). Other bounds on the number of iterations of the NM for FCO can be derived from the fact that for each iteration i, except the last one,

$$\frac{h(\delta_{i+1})}{h(\delta_i)} + \frac{g(\mathbf{x}^{(i+1)})}{g(\mathbf{x}^{(i)})} \leq 1, \tag{3}$$

which implies that sequence $(h(\delta_i) \cdot g(\mathbf{x}^{(i)}))$ decreases to 0 at a geometric rate. Using this fact one can show that the NM solves an integral linear FCOP in $O(\log(pU))$ iterations, and any linear FCOP in $O(p^2 \log^2 p)$ itera-



Fractional Combinatorial Optimization, Figure 1
The Newton method for FCO

tions [16,17]. The NM gives the asymptotically fastest known algorithm for the maximum mean-weight cut problem [16,17]. Its running time is $O(nm^2 \log n)$ for a graph with n nodes and m edges.

Megiddo's parametric search (MPS)

Let $\mathcal{A}_1$ be an algorithm as $\mathcal{A}_0$ above but with the following additional property: The value of each computed arithmetic expression on each possible execution path of algorithm $\mathcal{A}_1$ is a linear function of parameter δ. Such an algorithm $\mathcal{A}_1$ is called a *linear algorithm* for a parametric problem. MPS [13,14] solves a FCOP by following the computation of algorithm $\mathcal{A}_1$ for $\delta = \delta^*$. All values calculated during this computation are linear functions of (unknown) δ^* and are stored as such. Thus each comparison amounts to calculating the sign of the value of an expression $s - t\,\delta^*$, where s and t are known numbers, and can be resolved by running algorithm $\mathcal{A}_0$ for $\delta = s/t$ ($s/t \leq \delta^* \Leftrightarrow h(s/t) \geq 0$). If the running times of both $\mathcal{A}_1$ and $\mathcal{A}_0$ are at most T, then the overall running time of MPS is $O(T^2)$. If algorithm $\mathcal{A}_0$ runs in time T_0 and algorithm $\mathcal{A}_1$ is parallel and runs in time T_1 on P processors, then MPS can be implemented in such a way that the overall (sequential) running time is $O(T_1 P + T_0 T_1 \log P)$: At the kth (parallel) step of the computation of $\mathcal{A}_1$ for $\delta = \delta^*$, the required signs of P_k ($\leq P$) expressions $s_{k,j} - t_{k,j}\,\delta^*$, $j = 1, \ldots, P_k$, can be found by at most $\log P_k + 1$ executions of algorithm $\mathcal{A}_0$. The first execution is for δ equal to the median of the numbers $s_{k,j}/t_{k,j}$, and its result gives the signs of half of the expressions. MPS gives, for example, the asymptotically fastest known algorithms for the minimum ratio spanning-tree problem and the minimum cost-to-time ratio cycle problem [14]. Their running times are $O(m \log^2 n \log\log n)$ and $O(n^3 \log n \log\log n)$, respectively, for a graph with n nodes and m edges. An extension of MPS to cases when only approximate algorithms $\mathcal{A}_0$ are practical is proposed in [7] and applied there to the fractional 0–1 knapsack problem.

For some FCOPs, there are specialized algorithms which do not follow any of the above three generic methods. The most prominent examples are the $O(mn)$ [10] and $O(m\sqrt{n}\log(nU))$ [15] algorithms for the maximum mean cycle problem (the latter one is for the integral case). A detailed treatment of methods for FCO can be found in [17].

See also

- ▶ Bilevel Fractional Programming
- ▶ Combinatorial Matrix Analysis
- ▶ Combinatorial Optimization Algorithms in Resource Allocation Problems
- ▶ Combinatorial Optimization Games
- ▶ Complexity Classes in Optimization
- ▶ Complexity of Degeneracy
- ▶ Complexity of Gradients, Jacobians, and Hessians
- ▶ Complexity Theory
- ▶ Complexity Theory: Quadratic Programming
- ▶ Computational Complexity Theory
- ▶ Evolutionary Algorithms in Combinatorial Optimization
- ▶ Fractional Programming
- ▶ Information-based Complexity and Information-based Optimization
- ▶ Kolmogorov Complexity
- ▶ Mixed Integer Nonlinear Programming
- ▶ Multi-objective Combinatorial Optimization
- ▶ Neural Networks for Combinatorial Optimization
- ▶ NP-complete Problems and Proof Methodology
- ▶ Parallel Computing: Complexity Classes
- ▶ Quadratic Fractional Programming: Dinkelbach Method
- ▶ Replicator Dynamics in Combinatorial Optimization

References

1. Ahuja RK, Magnanti TL, Orlin JB (1993) Network flows: Theory, algorithms, and applications. Prentice-Hall, Englewood Cliffs, NJ
2. Chandrasekaran R (1977) Minimum ratio spanning trees. Networks 7:335–342
3. Dantzig GB, Blattner WO, Rao MR (1967) Finding a cycle in a graph with minimum cost to time ratio with application to a ship routing problem. In: Rosentiehl P (ed) Theory of Graphs. Gordon and Breach, New York, pp 77–83
4. Dinkelbach W (1967) On nonlinear fractional programming. Managem Sci 13:492–498
5. Ervolina TR, McCormick ST (1993) Two strongly polynomial cut cancelling algorithms for minimum cost network flow. Discrete Appl Math 46:133–165
6. Fox B (1969) Finding minimal cost-time ratio circuits. Oper Res 17:546–551
7. Hashizume S, Fukushima M, Katoh N, Ibaraki T (1987) Approximation algorithms for combinatorial fractional programming problems. Math Program 37:255–267

8. Ishii H, Ibaraki T, Mine H (1976) Fractional knapsack problems. Math Program 13:255–271
9. Iwano K, Misono S, Tezuka S, Fujishige S (1994) A new scaling algorithm for the maximum mean cut problem. Algorithmica 11:243–255
10. Karp RM (1978) A characterization of the minimum cycle mean in a digraph. Discret Math 23:309–311
11. Lawler EL (1967) Optimal cycles in doubly weighted directed linear graphs. In: Rosentiehl P (ed) Theory of Graphs. Gordon and Breach, New York, pp 209–213
12. Lawler EL (1976) Combinatorial optimization: networks and matroids. Holt, Rinehart and Winston, New York
13. Megiddo N (1979) Combinatorial optimization with rational objective functions. Math Oper Res 4:414–424
14. Megiddo N (1983) Applying parallel computation algorithms in the design of serial algorithms. J ACM 30:852–865
15. Orlin JB, Ahuja RK (1992) New scaling algorithms for assignment and minimum cycle mean problems. Math Program 54:41–56
16. Radzik T (1993) Parametric flows, weighted means of cuts, and fractional combinatorial optimization. In: Pardalos PM (ed) Complexity in Numerical Optimization. World Sci., Singapore, pp 351–386
17. Radzik T (1998) Fractional combinatorial optimization. In: Du D-Z, Pardalos PM (eds) Handbook Combinatorial Optim., vol 1. Kluwer, Dordrecht, pp 429–478

Fractional Programming

Hans Frenk[1], Siegfried Schaible[2]
[1] Econometric Institute, Erasmus University,
Rotterdam, The Netherlands
[2] A.G. Anderson Graduate School of Management,
University California, Riverside, USA

MSC2000: 90C32

Article Outline

Keywords
Classification
Single-ratio Fractional Programs
- Applications
- Economic Applications
- Maximization of Productivity
- Maximization of Return on Investment
- Maximization of Return/Risk
- Minimization of Cost/Time
- Maximization of Output/Input
- Noneconomic applications
- Indirect Applications
- Theoretical and Algorithmic Results
- Solving Problem (P) Directly
- Solving an Equivalent Problem (P_{eq})
- Solving a Dual Problem (D)
- Solving a Parametric Problem (P_q)
- Interior Point Algorithms
- Integer Fractional Programming

Maximization of the Smallest of Several Ratios
- Applications
- Theoretical and Algorithmic Results

Maximizing a Sum of Ratios
- Applications
- Theoretical and Algorithmic Results

Multi-objective Fractional Programming
Conclusion
See also
References

Keywords

Fractional programming; Single-ratio programs; Multiratio programs

In various applications of nonlinear programming a ratio of two functions is to be maximized or minimized. In other applications the objective function involves more than one ratio of functions. Ratio optimization problems are commonly called *fractional programs*.

One of the earliest fractional programs (though not called so) is an equilibrium model for an expanding economy introduced by J. von Neumann [50] in 1937. The model determines the growth rate as the maximum of the smallest of several output-input ratios. At a time when linear programming hardly existed, the author already proposed a duality theory for this nonconcave program.

However, apart from a few isolated papers like von Neumann's, a systematic study of fractional programming began much later. In 1962 A. Charnes and W.W. Cooper published their classical paper [19] in which they show that a linear fractional program can be reduced to a linear program with help of a nonlinear variable transformation.

The study of fractional programs with only one ratio has largely dominated the literature in this field until about 1980. Many of the results then known are presented in the first monograph on fractional programming by S. Schaible [57] (1978). Since then two other monographs solely devoted to fractional programming appeared, authored by B.D. Craven [23] and I.M. Stancu-Minasian [68]. Each of these includes a chapter on multi-ratio fractional programs.

Fractional programs have often been studied in the broader context of generalized convex programming [4]. Ratios of convex and concave functions as well as composites of such ratios are not convex in general, even in the case of linear ratios. But often they are generalized convex in some sense. From the beginning, fractional programming has benefited from advances in generalized convexity, and vice versa; see B. Martos [45]. This is demonstrated by the fact that the proceedings of each of the five international symposia on generalized convexity contain contributions on fractional programming; see [16,27,41,63,66]. Fractional programming overlaps also with global optimization. Several types of ratio optimization problems have local, nonglobal optima. For an extensive survey of fractional programming, see [60].

The survey [60] also contains the largest bibliography on fractional programming so far (1999). It has almost twelve-hundred entries. For a separate, rich bibliography see [68].

Clearly, fractional programming is a dynamic, growing area of research. It has been encouraging to observe that over the years research on theory and solution methods has increasingly more focused on those ratios which are of particular interest in applications. Since these are spread over a wide range of fields, surveys on fractional programming applications have been much needed. In the single-ratio case, a first detailed survey appeared in [57] and became a basis for [58,62] and for surveys by others. A more recent, detailed survey of single-ratio fractional programming applications is found in [68]. For the multi-ratio case, the surveys in [60,61,62] may be consulted. As various classes of fractional programs are presented below, the relevance of each class will be indicated.

Classification

Let f, g, h_k ($k = 1, \dots, m$) denote real-valued functions which are defined on a set C in the n-dimensional Euclidean space $\mathbf{R}^n$. Consider

$$q(x) = \frac{f(x)}{g(x)} \tag{1}$$

over the set

$$S = \{x \in C\colon\ h_k(x) \leq 0,\ k = 1, \dots, m\}, \tag{2}$$

assuming that $g(x) > 0$ on C. The nonlinear program

$$(\mathrm{P}) \quad \sup\{q(x)\colon\ x \in S\} \tag{3}$$

is called a (single-ratio) *fractional program*. In some applications more than one ratio appears in the objective function. Examples discussed in this article are

$$\sup\left\{\min_{1\leq i\leq p} q_i(x)\colon\ x \in S\right\} \tag{4}$$

and

$$\sup\left\{\sum_{i=1}^{p} q_i(x)\colon\ x \in S\right\}, \tag{5}$$

where $q_i(x)$ equals the ratio of the numerator $f_i(x)$ and the denominator $g_i(x)$ satisfying $g_i(x) > 0$ on C. Problem (4) is sometimes referred to as a *generalized fractional program* [62] while (5) is called a *sum-of-ratios fractional program*. Both problems (4) and (5) are related to the *multi-objective fractional program*

$$\max\left\{(q_1(x), \dots, q_p(x))\colon\ x \in S\right\}. \tag{6}$$

So far, the functions in the numerator and denominator were not specified. If f, g and h_k are affine functions (linear plus a constant) and C is the nonnegative orthant of $\mathbf{R}^n$, then (P) is called a *linear fractional program*. It is of the following form:

$$\sup\left\{\frac{c^\top x + \alpha}{d^\top x + \beta}\colon\ Ax \leq b, x \geq 0\right\}, \tag{7}$$

where $c, d \in \mathbf{R}^n$, $\alpha, \beta \in \mathbf{R}$, the superscript $\top$ denotes the transpose, A is an $m \times n$ matrix and $b \in \mathbf{R}^m$.

In generalization of a linear fractional program, we call (P) a *quadratic fractional program* if C is the nonnegative orthant, f, g are quadratic and the h_k are affine.

Problem (P) is said to be a *concave fractional program* if the numerator f is concave on C and g, h_k are convex on C, where C is a convex set. In addition, it is assumed that f is nonnegative on S if g is not affine. Note that the objective function of a concave fractional program (3) is generally not a concave function. Instead, it is composed of a concave and a convex function. Even under these restrictive concavity/convexity assumptions fractional programs are generally nonconcave programs.

The focus in fractional programming is the objective function and its ratio-structure. The feasible region is generally assumed to be convex or a convex polyhedron.

Single-ratio Fractional Programs

Consider the problem

$$\text{(P)} \quad \sup\{q(x)\colon \; x \in S\}\,, \tag{8}$$

where $q(x)$ equals the ratio of the numerator $f(x)$ and the denominator $g(x)$ with $g(x) > 0$ on C.

Applications

Fractional programs arise in management decision making as well as outside of it. They also occur sometimes indirectly in modeling where initially no ratio is involved. The purpose of the following overview is to demonstrate the diversity of problems which can be cast in the form of a single-ratio fractional program. A more comprehensive coverage which also includes the references for the models below is given in [60]. For other surveys of applications of (8) see [23,57,58,62,68].

Economic Applications

The efficiency of a system is sometimes characterized by a ratio of technical and/or economical terms. Maximizing the efficiency then leads to a fractional program. Some applications are given below.

Maximization of Productivity

P.C. Gilmore and R.E. Gomory [35] discuss a stock cutting problem in the paper industry for which under the given circumstances it is more appropriate to minimize the ratio of wasted and used amount of raw material rather than just minimizing the amount of wasted material. This stock cutting problem is formulated as a linear fractional program. In a case study, J.A. Hoskins and R. Blom [38] use fractional programming to optimize the allocation of warehouse personnel. The objective is to minimize the ratio of labor cost to the volume entering and leaving the warehouse.

Maximization of Return on Investment

In some resource allocation problems the ratio profit/capital or profit/revenue is to be maximized. A related objective is return per cost maximization. Resource allocation problems with this objective are discussed in more detail in [47]. In these models the term 'cost' may either be related to actual expenditure or may stand, for example, for the amount of pollution or the probability of disaster in nuclear energy production. Depending on the nature of the functions describing return, profit, cost or capital, different types of fractional programs are encountered. For example, if the price per unit depends linearly on the output and cost and capital are affine functions, then maximization of the return on investment gives rise to a concave quadratic fractional program (assuming linear constraints). In location analysis maximizing the profitability index (rate of return) is in certain situations preferred to maximizing the net present value, according to [5] and [8] and the cited references.

Maximization of Return/Risk

Some portfolio selection problems give rise to a concave nonquadratic fractional program of the form (11) below which expresses the maximization of the ratio of expected return and risk. For related concave and nonconcave fractional programs arising in financial planning see [60]. Markov decision processes may also lead to the maximization of the ratio of mean and standard deviation.

Minimization of Cost/Time

In several routing problems a cycle in a network is to be determined which minimizes the cost-to-time ratio or maximizes the profit-to-time ratio. Also the average cost objective used within the theory of stochastic regenerative processes [3] leads to the minimization of cost per unit time. A particular example occurring within this framework is the determination of the optimal ordering policy of classical periodic and continuous review single item inventory models, e. g., [31]. Another example of this framework are maintenance and replacement models. Here the ratio of the expected cost for inspection, maintenance and replacement and the expected time between two inspections is to be minimized, e. g., [6,30].

Maximization of Output/Input

Charnes, Cooper and E. Rhodes [22] use a linear fractional program as a model to evaluate the efficiency of decision making units (data envelopment analysis (DEA)). Given a collection of decision making units, the efficiency of each unit is obtained from the maximization of a ratio of weighted outputs and inputs subject to the condition that similar ratios for every decision making unit are less than or equal to unity. The variable weights are then the efficiency of each member relative to that of the others. For an extensive treatment of DEA see [21].

In the management literature there has been an increasing interest in optimizing relative terms such as relative profit. No longer are these terms merely used to monitor past economic behavior. Instead the optimization of rates is getting more attention in decision making processes for future projects; e. g., [5,37].

Noneconomic applications

In information theory the capacity of a communication channel can be defined as the maximal transmission rate over all probabilities. This is a concave nonquadratic fractional program. The eigenvalue problem in numerical mathematics can be reduced to the maximization of the Rayleigh quotient, and hence gives rise to a quadratic fractional program which is generally not concave. An example of a fractional program in physics is given by J.E. Falk [29]. He maximizes the signal-to-noise ratio of a spectral filter which is a concave quadratic fractional program.

Indirect Applications

There are a number of management science problems that indirectly give rise to a concave fractional program. A concave quadratic fractional program arises in location theory as the dual of a Euclidean multifacility minimax problem. In large scale mathematical programming, decomposition methods reduce the given linear program to a sequence of smaller problems. In some of these methods the subproblems are linear fractional programs. The ratio originates in the minimum-ratio rule of the simplex method.

Fractional programs are also met indirectly in stochastic programming, as first shown in [20] and [13]. This will be illustrated by two models below [57,68]. First, consider the stochastic mathematical program:

$$\max\left\{a^\top x\colon\ x\in S\right\}, \tag{9}$$

where the coefficient vector a is a random vector with a multivariate normal distribution and S is a (deterministic) convex feasible region. It is assumed that the decision maker replaces (9) by a decision problem

$$\max\left\{\mathsf{P}\{a^\top x\geq k\}\colon\ x\in S\right\}, \tag{10}$$

i. e., he wants to maximize the probability that the random variable $a^\top x$ attains at least a prescribed level k. Then (9) reduces to

$$\max\left\{\frac{e^\top x-k}{\sqrt{x^\top Vx}}\colon\ x\in S\right\}, \tag{11}$$

where e is the mean vector of the random vector a and V its variance-covariance matrix. Hence the maximum probability model of the concave program (9) gives rise to a concave fractional program. If in (9) the linear objective function is replaced by other types of nonlinear functions, then the maxi- mum probability model leads to various other concave fractional programs as demonstrated in [57,68].

Consider a second stochastic program

$$\max\{f_0(x)+\theta f_1(x)\colon\ x\in S\}, \tag{12}$$

where f_0, f_1 are concave functions on the convex feasible region S, $f_1>0$ and θ is a random variable with a continuous cumulative distribution function. Then the maximum probability model for (12) gives rise to the fractional program

$$\max\left\{\frac{f_0(x)-k}{f_1(x)}\colon\ x\in S\right\}. \tag{13}$$

For a linear program (12) the deterministic equivalent (13) becomes a linear fractional program. If f_0 is concave and f_1 linear, then (13) is still a concave fractional program. However, if f_1 is also a (nonlinear) concave function, then (13) is no longer a concave fractional program. Obviously a quadratic program (12) reduces to a quadratic fractional program. For more details on (12), (13) see [57,68].

Stochastic programs (9) and (12) are met in a wide variety of planning problems. Whenever the maximum probability model is used as a deterministic equivalent, such decision problems lead to a fractional program of one type or another. Hence, fractional programs are encountered indirectly in many different applications of mathematical programming, although initially the objective function is not a ratio.

With the recent advent of various interior point methods for linear programming problems fractional programming has been given more attention as well. For instance, K.M. Anstreicher [2] showed that Karmarkar's projective algorithm is fundamentally an algorithm for linear fractional programming on a simplex.

M. Gaudioso and M.F. Monaco [34] use quadratic fractional programs as subproblems in an algorithm for convex nondifferentiable programs. These arise as duals of search direction subproblems.

Theoretical and Algorithmic Results

Most of the algorithms known so far solve linear, or more generally, concave fractional programs (8). At least five different strategies are found in the literature and are reviewed below.

Solving Problem (P) Directly

Concave (linear) fractional programs share some important properties with concave (linear) programs, due to the generalized concavity (and in addition, generalized convexity in the linear case) of the objective function $q(x) = f(x)/g(x)$ [4,45]:

1) a local maximum is a global maximum;
2) a maximum is unique if either the numerator is strictly concave or the denominator is strictly convex;
3) a solution of the Karush–Kuhn–Tucker optimality conditions is a maximum, assuming f, g, h_k are differentiable on the open set C;
4) a maximum is attained at an extreme point of the convex polyhedron S of a linear fractional program (provided an optimal solution exists).

Because of the properties 1) and 3), it is possible to solve concave fractional programs by several of the standard concave programming algorithms. Indeed, it was shown that certain concave programming methods can be applied to programs with a quasiconcave objective function [45]; for example, the Frank–Wolfe linearization method [23,45]. M. Boncompte and J.E. Martinez-Legaz [14] proposed a cutting plane method for concave fractional programs, based on the upper subdifferentiability of the objective function. If (P) is a linear fractional program, then property 4) can be used to calculate a maximum $\overline{x}$ by determining a finite sequence of extreme points x_i of S with increasing values $q(x_i)$ converging to $\overline{x}$. Thus a simple simplex-like procedure can solve linear fractional programs [45].

Solving an Equivalent Problem (P_{eq})

Some of the concave programming algorithms are not suitable for generalized concave programs [45]. Thus the choice of concave programming algorithms to solve concave fractional programs directly is limited. However, it can be shown that every concave fractional program is transformable into a concave program: the variable transformation

$$y = \frac{x}{g(x)} \quad \text{and} \quad t = \frac{1}{g(x)} \tag{14}$$

reduces (P) to

$$(\mathrm{P}_{\mathrm{eq}}) \quad \sup \left\{ tf\left(\frac{y}{t}\right) : (y, t) \in \widetilde{S} \right\} \tag{15}$$

with the region $\widetilde{S}$ represented by the relations

$$th_k\left(\frac{y}{t}\right) \leq 0, \quad tg\left(\frac{y}{t}\right) \leq 1, \quad \frac{y}{t} \in C, \quad t > 0, \tag{16}$$

and this is a concave program [55]. If $(\overline{y}, \overline{t})$ is an optimal solution of (P_{eq}), then $\overline{x} = \overline{y}/\overline{t}$ is an optimal solution of (P). Such a transformation was originally suggested by Charnes and Cooper [19] who showed that with help of (14) a linear fractional program can be reduced to a linear program. Because of the transformability into a concave program, concave fractional programs can indirectly be solved by *any* concave programming method, applying the algorithm to the equivalent program (P_{eq}). Hence through transformation (14) one gains access to *all* convave programming algorithms.

To solve (P_{eq}) rather than (P) may be particularly appropriate when the numerator f and the denominator g have a certain algebraic structure. For example, the maximum probability model (11) or certain portfolio selection models have an affine numerator and a denominator which is the square root of a convex quadratic form. In this case (P_{eq}) reduces to a concave quadratic program, and hence (P) can directly be solved by one of the standard quadratic programming techniques [59]. In the special case of a linear fractional program (7) transformation (14) yields the linear program

$$\sup\left\{c^{\top}y+\alpha t\colon\ (y,t)\in\widehat{S}\right\} \tag{17}$$

with the feasible region $\widehat{S}$ represented by the relations

$$Ay-bt\le 0,\quad d^{\top}y+\beta t=1,\quad y\ge 0,\quad t>0.$$

Hence (7) can be solved by the simplex method [19]. For a comparison with other linear fractional programming methods see [60].

Solving a Dual Problem (D)

One of the disadvantages of solving (P) directly is that duality concepts of concave programming cannot be used since basic duality relations are no longer valid for these nonconcave programs. However, transformation (14) enables us to gain access to concave programming duality. Thus a dual fractional program can be defined as one of the classical duals of the equivalent concave program (P_{eq}) [55]. For instance, the Lagrangian dual of (P_{eq}) gives rise to the dual fractional program

$$\text{(D)}\quad \inf\left\{\sup_{x\in C}\left\{\frac{f(x)-u^{\top}h(x)}{g(x)}\colon\ u\ge 0\right\}\right\}, \tag{18}$$

where $h=(h_1,\dots,h_m)^{\mathsf{T}}$. As in concave programming, several duality relationships can be established between (P) and (D) [55].

Various duals have been suggested in different approaches [57,59]. However, not much effort has been devoted to algorithmically using duality. In [56] the dual is used to calculate bounds in an iterative procedure for concave fractional programs. Much remains to be done to take full advantage of fractional programming duality in algorithms.

For the dual (D) to be a computationally attractive alternative to (P) or (P_{eq}), the fractional program (P) should have a certain amount of algebraic structure in f, g and h_k. Otherwise it may well be easier to solve (P) rather than a dual of (P). If (P) is a concave quadratic fractional program with an affine denominator, then the dual can be written as a linear program with one additional concave quadratic constraint [59,65].

One advantage of a dual method is that in addition to an optimal solution of (P) also the sensitivity of the maximal value of $q(x)$ with regard to right-hand side changes can be calculated. The dual variables $\overline{u}_i$ in an optimal solution turn out to be propertional to the marginal values of $q(x)$ at $\overline{x}$ [57,58,59]. Sensitivity and parametric analysis for fractional programming has been extensively discussed; see [18,23,57,58] and the cited references.

Solving a Parametric Problem (P_q)

There is a rich class of algorithms based on the following parametric problem associated with (P):

$$(P_q)\quad \max\{f(x)-qg(x)\colon\ x\in S\}, \tag{19}$$

where $q\in\mathbf{R}$ is a parameter. The program (P_q) is sometimes numerically more tractable than the program (P). For example, (P_q) is a parametric quadratic (linear) program if (P) is a quadratic (linear) fractional program, and (P_q) is a parametric concave program if (P) is a concave fractional program. M. Sniedovich [67] analyzed the relationship between (P_q) and classical optimization techniques applied to (P).

In the following it is assumed that S is compact and f, g are continuous on S. Let $F(q)$ denote the optimal value of the objective function of (P_q). $F(q)$ is a strictly decreasing, convex function which has a unique zero $q=\overline{q}$. An optimal solution $\overline{x}$ of ($P_{\overline{q}}$)) is also an optimal solution of (P) with $\overline{q}=f(\overline{x})/g(\overline{x})$. Thus solving (P) is equivalent to finding the unique root of the nonlinear equation $F(q)=0$. With the properties of $F(q)$ in view, T. Ibaraki [39] applied various classical search techniques to calculate the zero $q=\overline{q}$. These interval-type algorithms generate a succession of intervals with decreasing amplitude containing $q=\overline{q}$. Computational results are reported in [39,62]. The application of Newton's method is commonly referred to as the algorithm by W. Dinkelbach, who first proposed such a procedure [28]. Its equivalence to Newton's method was seen later by Ibaraki. A very efficient version of Dinkelbach's method was suggested in [51] improving an earlier variant in [56].

Interior Point Algorithms

In addition to the four more classical strategies above, recently new techniques have emerged which are of the interior point type. The first such method, developed for linear fractional programs, is due to Anstreicher [2]. In 1994, R.W. Freund and F. Jarre [32] proposed a method for concave fractional programs. A polynomial convergence is established and some numerical results are reported.

Most of the computational work in single-ratio fractional programming compares and tests algorithms that use the parametric program ($\mathbf{P}_q$). Much more work is needed to compare computationally the various four approaches above with each other and with the very recent polynomial time interior point methods. Also new methods need to be developed for certain nonconcave fractional programs arising in applications; e. g., [59].

This section on single-ratio fractional programming concludes with a brief discussion of integer fractional programming. In some of the economic applications above the variables are restricted to be integers, if indivisible goods are involved. Also, a number of combinatorial fractional programs with 0–1 variables are of interest; for instance fractional location problems [5].

Integer Fractional Programming

This is an important, but somewhat neglected field within fractional programming. In [5] A.I. Barros gives an overview of some of the advances. Here the parametric procedure by N. Megiddo [46] stands out particularly. T. Radzik [53] provided a detailed survey of the advances in *combinatorial fractional programming*. The survey includes many of his own complexity results on Dinkelbach's and Megiddo's parametric procedures. Among others, Radzik shows that Dinkelbach's algorithm solves a combinatorial linear fractional program in a strongly polynomial number of iterations, regardless of the constraint structure. Some of the results in [53] are specialized to cases such as the problem of profit-to-time cycles and maximum mean-weight cuts. In the same survey also open problems in combinatorial fractional programming are identified.

Leaving the single-ratio case now, the three multi-ratio fractional programs in (4), (5) and (6) will be addressed below. Among these, so far best researched is the following.

Maximization of the Smallest of Several Ratios

Consider the Problem

$$\sup\left\{\min_{1\leq i\leq p} q_i(x)\colon\ x\in S\right\}, \tag{20}$$

where $q_i(x) = f_i(x)/g_i(x)$ and

$$S = \{x\in C\colon\ h_k(x)\leq 0,\ k=1,\dots,m\}.$$

It is assumed that $C \subseteq \mathbf{R}^n$ is nonempty, convex and h_k are real-valued convex functions on C. Before analyzing (20), some applications of this model are outlined.

Applications

In mathematical economics problem (20) may arise when the growth rate of an expanding economy is determined [50]:

$$\text{growth rate} = \max_x\left(\min_{1\leq i\leq p}\frac{\text{output}_i(x)}{\text{input}_i(x)}\right), \tag{21}$$

where x denotes a feasible production plan of the economy. In management science simultaneous maximization of rates such as those discussed earlier can lead to (20). This is so if either in a worst-case approach the model

$$\min_{1\leq i\leq p}\frac{f_i(x)}{g_i(x)} \to \sup \tag{22}$$

is used or with the help of prescribed ratio goals r_i

$$\max_{1\leq i\leq p}\left|\frac{f_i(x)}{g_i(x)} - r_i\right| \to \inf \tag{23}$$

is employed. In both cases essentially a *max-min fractional program* (20) is to be solved. Examples of the second approach are found in financial planning with different financial ratios or in the allocation of funds under equity considerations. Furthermore (20) was recently used in location analysis; see [5] for details. A third area of application of model (20) is numerical mathematics. Given the values F_i of a function $F(t)$ in finitely many points t_i of an interval for which an approximating ratio of two polynomials $N(t, x_1)$ and $D(t, x_2)$ with coefficient vectors x_1, x_2 is sought. If the best approximation is defined in the sense of the L_∞-norm, then the following problem is to be solved:

$$\max_i\left|\frac{N(t_i,x_1)}{D(t_i,x_2)} - F_i\right| \to \inf \tag{24}$$

with variables x_1, x_2. Like (23), this problem can be reduced to a max-min fractional program (20).

Theoretical and Algorithmic Results

Several authors, including von Neumann [50], have introduced dual programs for problem (20) employing different approaches; see [60]. In most duality approaches the following assumptions are made: C is nonempty, convex and compact, $-f_i$, g_i, h_k are lower semicontinuous, $-f_i$, g_i, h_k are convex, g_i are positive on C, f_i are nonnegative on S, if at least one g_i is not affine, and the feasible region S is nonempty. Let $F = (f_1, \ldots, f_p)^\top$, $G = (g_1, \ldots, g_p)^\top$ and $h = (h_1, \ldots, h_m)^\top$. The following dual is derived in [40] with help of the Farkas lemma (cf. ► Farkas lemma; ► Farkas lemma: Generalizations):

$$\inf_{\substack{u \geq 0, \\ v \geq 0, v \neq 0}} \left\{ \sup_{x \in C} \left\{ \frac{v^\top F(x) - u^\top h(x)}{v^\top G(x)} \right\} \right\} . \qquad (25)$$

Under the assumptions above the optimal values in (20) and (25) coincide, and duality relations much like those in concave and linear programming hold [40]; see also [25].

The primal max-min program (20) is associated with a dual *min-max fractional program* (25). Such a symmetry is not obvious in single-ratio fractional programming duality theory; see (18). Symmetry between the primal and dual exists also in the following sense: in both problems a local optimum is a global optimum. This follows from the fact that the primal objective function is semistrictly quasiconcave and the dual objective function is semistrictly quasiconvex [4]. The dual objective function usually involves infinitely many ratios in contrast to the primal one. However, this asymmetry disappears in case of a linear problem (20) where f_i, g_i, h_k are affine and C is the (unbounded) nonnegative orthant of $\mathbf{R}^n$. Then only finitely many ratios need to be considered in the dual objective function. In the linear case it can further be shown that in addition to the usual complementary slackness between variables in one problem and constraints in the other one, complementary slackness also exists between certain variables in one and ratios in the other one [25]. Hence in the linear case of (20) there exist complete symmetry as well as a close relationship between the primal and the dual fractional program.

Regarding solution methods for (20), an extension of Dinkelbach's algorithm to (20) was introduced by J.P. Crouzeix, J.A. Ferland and Schaible in [26]. It proved to have attractive convergence properties and became the starting point for the design of similar methods surveyed in [24]. Several of these interval-type methods have been compared and tested. M. Gugat [36] proposed a fast interval-type algorithm for (20) which always converges superlinearly. Boncompte and Martinez–Legaz [14] used a cutting plane approach, originally suggested in [52] for a more general class of quasiconcave problems, employing upper subdifferentiability of the objective function in (20). A computational comparison with the Dinkelbach-type method in [26] is carried out too. A different cutting plane method incorporating the ideas of [52] and [67], again for a more general class of problems than generalized fractional programs, is discussed in [7]. In case of problem (20) the method in [7] reduces to the Dinkelbach-type method in [26]. Thus the latter can also be viewed as a cutting plane method.

Most of the algorithms above solve the primal problem (20). In the linear case the Dinkelbach-type algorithm in [26] can also be applied to the dual because of symmetry between (P) and (D). Recently a 'dual' algorithm for (20) was proposed in the nonlinear case [10]. It can be viewed as an extension of the Dinkelbach-type algorithm in [26] applied to the dual of a generalized linear fractional program. In [9] a new dual of (20) was proposed as well as an efficient method to solve it. Less restrictive assumptions ensure superlinear convergence of this new 'dual' algorithm. An extensive computational comparison of the Dinkelbach-type method in [26] with the two dual methods was performed by Barros, J.B.G. Frenk, Schaible and S. Zhang; see [5,9,10]. The test problems involve quadratic ratios.

Freund and Jarre [33] proposed an interior point method for solving (20) which extends their method in [32] for single-ratio problems. Furthermore, A.S. Nemirovsky and Yu.E. Nesterov [48,49] developed several interior point algorithms for (20) which converge in polynomial time. The studies above contain thorough complexity analyses. Summarizing, one can say that max-min fractional programs have been researched quite extensively.

Maximizing a Sum of Ratios

Consider the multi-ratio fractional program

$$\sup\left\{\sum_{i=1}^{p} q_i(x):\ x\in S\right\}\ , \tag{26}$$

where $q_i(x) = f_i(x)/g_i(x)$, $g_i(x) > 0$.

Applications

Model (26) arises naturally in decision making when several rates are to be optimized simultaneously and a compromise is sought which optimizes a weighted sum of these rates. In light of the applications in the single-ratio case, numerators and denominators may represent profit, cost, capital, risk or time, for example. Model (26) also includes the case where some ratios are not proper quotients, i. e., $g_i(x) = 1$. This describes situations where a compromise is sought between absolute and relative terms like profit and return on investment (profit/capital) or return and return/risk. Additional applications of (26) are surveyed in [61]. To mention a few, Y. Almogy and O. Levin [1] analyze a multistage stochastic shipping problem and show that a deterministic equivalent of this stochastic problem leads to (26). M.R. Rao [54] discusses various models in cluster analysis. The problem of optimal partitioning of a given set of entities into a number of mutually exclusive and exhaustive groups (clusters) gives rise to various mathematical programming problems, depending on which optimality criterion is used. If the objective is to minimize the sum of the average squared distances within groups, then a minimum of a sum of ratios is to be determined. H. Konno and M. Inori [42] formulated a bond portfolio optimization problem in the form (26).

Theoretical and Algorithmic Results

As seen earlier, the case of ratios of concave and convex functions is of particular interest in applications. Fortunately, it lends itself to a relatively easy analysis of models (8) and (20). A local maximum is a global one, duality relations can be established and several efficient solution techniques are available. Unfortunately, for the sum-of-ratios problem none of this is true any longer if in (26) all ratios $f_i(x)/g_i(x)$ are quotients of concave and convex functions. In particular, a local maximum is usually not a global one, even in the case of linear ratios. More often the objective function is not quasiconcave. I.A. Bykadorov [15] studied certain generalized concavity properties of sums of linear ratios and, more generally, of sums of ratios of polynomials. Only some limited theoretical results are known for the sum of concave ratios; see [23] and the surveys [60,61]. In the case of linear ratios, C.H. Scott and T.R. Jefferson [64] proposed a duality concept for (26) using geometric programming duality.

Given the small theoretical basis, it is not surprising that algorithmic advances have been rather limited too. Several strategies have been proposed and are surveyed in [61]. The best tested method can be found in [43]. Separating numerators and denominators with help of additional variables, problem (26) is embedded into a higher-dimensional space with a concave objective function. A global minimum is then found through approximation techniques. Computational experience with the related multiplicative program in [43] shows that the method works quite well for up to about four terms. However, for more terms in the sum it looses its efficiency fast. Much work is still necessary to develop efficient algorithms for (26), even in the case of linear ratios.

Multi-objective Fractional Programming

The problem of simultaneously maximizing several ratios leads to a *multi-objective fractional program*

$$\max\left\{(q_1(x),\dots,q_p(x)):\ x\in S\right\}, \tag{27}$$

where $q_i(x) = f_i(x)/g_i(x)$, $g_i(x) > 0$. Such a model arises when in contrast to the previous two models (20) and (26) a unifying objective is not considered. Instead, the decision-maker is to be provided with some or all efficient (Pareto optimal) alternatives. These are feasible solutions such that no ratio can be further increased without decreasing at least one of the other ratios. Applications, for instance in financial planning or production planning, can easily be envisioned in light of the applications of fractional programming described earlier; see also [44,60,68] and references therein. In case of concave ratios problem (27) can be seen as a special case of a semistrictly quasiconcave multi-objective programming problem; e. g., [17] and articles in [16,27,41,63,66]. Duality for multi-objective fractional programs has been studied by several authors, usually for concave or linear ratios; see [11,68]. For such problems also equivalences to multi-objective programs without ratios have been established [68]. These are formed with help of the numerator and denominator functions.

Another topic, important from a theoretical and algorithmic point of view, is the question whether the set *E* of efficient (Pareto optimal) solutions is connected. Only partial answers were available until very recently [60]. Meanwhile connectedness has been shown for continuous concave fractions over a compact convex feasible region. This is a consequence of a more general result in [12] for semistrictly quasiconcave objective functions. Several solution methods for the calculation of (weakly, proper) efficient solutions have been proposed for linear and concave ratios; see [44,68] and cited references. It is noted that the calculation of *E* may simplify the solution of the difficult sum-of-ratios problem [60] since an optimal solution of (26) is an efficient solution. Such an approach seems to be particularly promising in case of two ratios. In summary, some good progress has been made in the analysis of concave multi-objective fractional programs. However more work is needed.

Conclusion

Many interesting problems inside and outside management decision making gives rise to the optimization of one or several ratios. Much effort has been devoted to the analysis of such nonconcave programs. However, the theoretical basis is still not broad enough, especially for sum-of-ratios problems and, to a lesser extend, for multi-objective fractional programs. The computational experience with fractional programs is also quite limited. Major progress has been made for concave single-ratio and max-min fractional programs. But much more work is necessary for the other fractional programs of interest in applications.

See also

- ► Bilevel Fractional Programming
- ► Fractional Combinatorial Optimization
- ► Quadratic Fractional Programming: Dinkelbach Method

References

1. Almogy Y, Levin O (1970) Parametric analysis of a multistage stochastic shipping problem. In: Lawrence J (ed) Operational Research '69. Tavistock Publ., London, pp 359–370
2. Anstreicher KM (1986) A monotonic projective algorithm for fractional linear programming. Algorithmica, 1(4):483–498
3. Asmussen S (1987) Applied probability and queues. Wiley, New York
4. Avriel M, Diewert WE, Schaible S, Zang I (1988) Generalized concavity. Plenum, New York
5. Barros AI (1998) Discrete and fractional programming. Kluwer, Dordrecht
6. Barros AI, Dekker R, Frenk JBG, Weeren S van (1997) Optimizing a general optimal replacement model by fractional programming techniques. J Global Optim 10:405–423
7. Barros AI, Frenk JBG (1995) Generalized fractional programming and cutting plane algorithms. J Optim Th Appl 87(1):103–120
8. Barros AI, Frenk JBG, Gromicho J (1997) Fractional location problems. Location Sci 5(1):47–58
9. Barros AI, Frenk JBG, Schaible S, Zhang S (1996) A new algorithm for generalized fractional programs. Math Program 72(2):147–175
10. Barros AI, Frenk JBG, Schaible S, Zhang S (1996) Using duality to solve generalized fractional programming problems. J Global Optim 8:139–170
11. Bector CR, Chandra S, Singh C (1990) Duality in multiobjective fractional programming. In: Cambini A, et al (eds) Generalized Convexity and Fractional Programming with Economic Applications. Lecturer Notes Economics and Math Systems, vol 345. Springer, Berlin, pp 232–241
12. Benoist J (1998) Connectedness of the efficient set for strictly quasiconcave sets for strictly quasiconcave sets. J Optim Th Appl, 96:627–654
13. Bereanu B (1965) Decision regions and minimum risk solutions in linear programming. In: Prekopa A (ed) Coll. Applications of Mathematics to Economics. Hungarian Acad. Sci., Budapest, pp 37–42
14. Boncompte M, Martinez-Legaz JE (1991) Fractional programming by lower subdifferentiability techniques. J Optim Th Appl 68(1):95–116
15. Bykadorov IA (1994) On quasiconvexity in fractional programming. In: Komlosi S, Rapcsak T, Schaible S (eds) Generalized Convexity, Proc. Pécs/Hungary, 1992. Lecture Notes Economics and Math Systems, vol 405. Springer, Berlin, pp 281–293
16. Cambini A, Castagnoli E, Martein L, Mazzoleni P, Schaible S (eds) (1990) Generalized convexity and fractional programming with economic applications, Proc. 3rd Internat. Workshop at Pisa, Italy, 1988. Lecture Notes Economics and Math Systems, vol 345. Springer, Berlin

17. Cambini A, Martein L (1998) Generalized concavity in multiobjective programming. In: Crouzeix JP, Martinez-Legaz JE, Volle M (eds) Generalized Convexity, Generalized Monotonicity. Kluwer, Dordrecht, pp 453–467
18. Cambini A, Schaible S, Sodini C (1993) Parametric linear fractional programming for an unbounded feasible region. J Global Optim 3:157–169
19. Charnes A, Cooper WW (1962) Programming with linear fractional functionals. Naval Res Logist Quart, 9:181–186
20. Charnes A, Cooper WW (1963) Deterministic equivalents for optimizing and satisficing under chance constraints. Oper Res, 11:18–39
21. Charnes A, Cooper WW, Lewin AY, Seiford LM (eds) (1994) Data envelopment analysis: Theory, methodology and applications. Kluwer, Dordrecht
22. Charnes A, Cooper WW, Rhodes E (1978) Measuring the efficiency of decision making units. Europ J Oper Res 2:429–444
23. Craven BD (1988) Fractional programming. Sigma Ser Appl Math, vol 4. Heldermann, Berlin
24. Crouzeix JP, Ferland JA (1991) Algorithms for generalized fractional programming. Math Program 52:191–207
25. Crouzeix JP, Ferland JA, Schaible S (1983) Duality in generalized linear fractional programming. Math Program 27(3):342–354
26. Crouzeix JP, Ferland JA, Schaible S (1985) An algorithm for generalized fractional programs. J Optim Th Appl 47(1):35–49
27. Crouzeix JP, Martinez-Legaz JE, Volle M (eds) (1998) Generalized convexity, generalized monotonicity: Recent results, Proc. 5th Internat. Symp. Kluwer, Dordrecht
28. Dinkelbach W (1967) On nonlinear fractional programming. Managem Sci 13:492–498
29. Falk JE (1969) Maximization of signal-to-noise ratio in an optical filter. SIAM J Appl Math 7:582–592
30. Frenk JBG, Dekker R, Kleijn M (1997) On the marginal cost approach in maintenance. J Optim Th Appl 94(3):771–781
31. Frenk JBG, Kleijn MJ (1998) On regenerative processes and inventory control. Pure Appl Math 9:61–94
32. Freund RW, Jarre F (1994) An interior-point method for fractional programs with convex constraints. Math Program 67(3):407–440
33. Freund RW, Jarre F (1995) An interior-point method for multifractional programs with convex constraints. J Optim Th Appl 85:125–161
34. Gaudioso M, Monaco MF (1991) Quadratic approximations in convex non-differentiable optimization. SIAM J Control Optim 29:58–70
35. Gilmore PC, Gomory RE (1963) A linear programming approach to the stock cutting problem – Part II. Oper Res 11:863–888
36. Gugat M (1996) A fast algorithm for a class of generalized fractional programs. Managem Sci 42(10):1493–1499
37. Heinen E (1971) Grundlagen betriebswirtschaftlicher Entscheidungen, Das Zielsystem der Unternehmung, 2nd edn. Gabler, Wiesbaden
38. Hoskins JA, Blom R (1984) Optimal allocation of warehouse personnel: A case study using fractional programming. Focus (UK) 3(2):13–21
39. Ibaraki T (1983) Parametric approaches in fractional programs. Math Program 26(3):345–362
40. Jagannathan R, Schaible S (1983) Duality in generalised fractional programming via Farkas' lemma. J Optim Th Appl 41(3):417–424
41. Komlosi S, Rapcsak T, Schaible S (eds) (1994) Generalized convexity, Proc. 4th Internat. Workshop at Pécs/Hungary, 1992. Lecture Notes Economics and Math Systems, vol 405. Springer, Berlin
42. Konno H, Inori M (1989) Bond porfolio optimiza tion by bilinear fractional programming. J Oper Res Soc Japan 32(2):143–158
43. Konno H, Kuno T, Yajima Y (1994) Global minimization of a generalized convex multiplicative function. J Global Optim 4:47–62
44. Kornbluth JSH, Steuer RE (1981) Multiple objective linear fractional programming. Managem Sci 27:1024–1039
45. Martos B (1975) Nonlinear programming, theory and methods. North-Holland, Amsterdam
46. Megiddo N (1979) Combinatorial optimization with rational objective functions. Math Oper Res 4:414–424
47. Mjelde KM (1983) Methods of the allocation of limited resources. Wiley, New York
48. Nemirovski AS (1997) A long-step method of analytical centers for fractional programs. Math Program 77:191–224
49. Nesterov YuE, Nemirovski AS (1995) An interior point method for generalized linear-fractional problems. Math Program 69:117–204
50. Von Neumann J (1937) "Über ein ökonomisches Gleichungssystem und eine Verallgemeinerung des Brouwerschen Fixpunktsatzes. In: Menger K (ed) Ergebn. eines math. Kolloqiums, vol 8, pp 73–83
51. Pardalos PM, Phillips AT (1991) Global optimization of fractional programs. J Global Optim 1:173–182
52. Plastria F (1985) Lower subdifferentiable functions and their minimization by cutting planes. J Optim Th Appl Appl. 46:37–53
53. Radzik T (1998) Fractional combinatorial optimization. In: Du D-Z, Pardalos PM (eds) Handbook Combinatorial Optim., vol 1. Kluwer, Dordrecht, pp 429–478
54. Rao MR (1971) Cluster analysis and mathematical programming. J Amer Statist Assoc 66:622–626
55. Schaible S (1976) Fractional programming: I, Duality. Managem Sci 22(8):858–867
56. Schaible S (1976) Fractional programming: II, On Dinkelbach's algorithm. Managem Sci 22(8):868–873
57. Schaible S (1978) Analyse und Anwendungen von Quotientenprogrammen, Ein Beitrag zur Planung mit Hilfe der

nichtlinearen Programmierung, Mathematical Systems in Economics, vol 42. Anton Hain Verlag, Meisenheim

58. Schaible S (1981) Fractional programming: Applications and algorithms. Europ J Oper Res 7:111–120
59. Schaible S (1983) Fractional programming. Z Oper Res, Ser A 27(1):39–54
60. Schaible S (1995) Fractional programming. In: Horst R, Pardalos PM (eds) Handbook Global Optim. Kluwer, Dordrecht, 495–608
61. Schaible S (1996) Fractional programming with sums of ratios. In: Castagnoli E, Giorgi G (eds) Scalar and Vector Optimization in Economic and Financial Problems, Proc. Workshop Milan, 1995. Elioprint, Perugia, 163–175
62. Schaible S, Ibaraki T (1983) Fractional programming (invited review). Europ J Oper Res 12(4):325–338
63. Schaible S, Ziemba WT (eds) (1981) Generalized concavity in optimization and economics. Proc NATO Advanced Study Inst Vancouver, Canada 1980. Acad. Press, New York
64. Scott CH, Jefferson TR (1998) Duality for a nonconvex sum of ratios. J Optim Th Appl 98:151–159
65. Scott CH, Jefferson TR, Frenk JBG (1998) A duality theory for a class of generalized fractional programs. J Global Optim 12:239–245
66. Singh C, Dass BK (eds) (1989) Continuous-time, fractional and multiobjective programming. Proc. Internat. Conf. St. Lawrence Univ., Canton, NY, 1986, Analytic Publ., Delhi, also: J. Inform. Optim. Sci. 10(1)
67. Sniedovich M (1988) Fractional programming revisited. Europ J Oper Res 33(3):334–341
68. Stancu-Minasian IM (1997) Fractional programming: Theory, methods and applications. Kluwer, Dordrecht

Fractional Zero-One Programming

OLEG PROKOPYEV
Department of Industrial Engineering,
University of Pittsburgh, Pittsburgh, USA

Article Outline

Introduction
Applications
Complexity Issues
Mixed Integer Reformulation
Solution Techniques
References

Introduction

One of the classes of 0-1 optimization problems is the maximization (or minimization) of a sum of ratios of linear 0-1 functions:

$$\max_{x\in\{0,1\}^n} f(x) = \sum_{j=1}^{m} \frac{a_{j0} + a_j^T x}{b_{j0} + b_j^T x}\,, \tag{1}$$

$$\text{s.t. } Dx \le c\,, \tag{2}$$

where $a_j \in \mathbb{R}^n$, $b_j \in \mathbb{R}^n$, $a_{j0} \in \mathbb{R}$, $b_{j0} \in \mathbb{R}$, $D \in \mathbb{R}^{k\times n}$ and $c \in \mathbb{R}^k$. Problem (1)–(2) is referred to as *fractional 0-1 programming problem* [21], or *hyperbolic 0-1 programming problem* [1,20].

Note that if for some j and x in the feasible region (2) the term $b_{j0} + b_j^T x$ is equal to zero, then problem (1)–(2) may not have a finite optimum. Therefore, it is usually assumed that

$$b_{j0} + b_j^T x \neq 0\,,\ \text{for all } x \in \{0,1\}^n \text{ and } j = 1,\dots,m\,. \tag{3}$$

Furthermore, sometimes we can make a stricter assumption and require that all denominators in (1) are positive, i. e.,

$$b_{j0} + b_j^T x > 0\,,\ \text{for all } x \in \{0,1\}^n \text{ and } j = 1,\dots,m\,. \tag{4}$$

A special simplified class of (1)–(2) is the so-called *single-ratio fractional (hyperbolic) 0-1 programming problem*:

$$\max_{x\in\{0,1\}^n} f(x) = \frac{a_0 + \sum_{i=1}^n a_i x_i}{b_0 + \sum_{i=1}^n b_i x_i}\,. \tag{5}$$

Problem (1) can be generalized if instead of linear 0-1 functions we consider 0-1 polynomials:

$$\max_{x\in\{0,1\}^n} f(x) = \sum_j \frac{a_{j0} + \sum_{S\in A_j} a_{jS} \prod_{i\in S} x_i}{b_{j0} + \sum_{T\in B_j} b_{jT} \prod_{i\in T} x_i}\,, \tag{6}$$

where A_j and B_j are families of subsets of $\{1,2,\dots,n\}$.

In general case, problems of type (1), (5) and (6) can be considered subject to various 0-1 linear and nonlinear constraints. A specific class of fractional 0-1 programming problems, where fractional terms appear not in the objective function, but in the set of constraints, is discussed in [2]:

$$\max_{x\in\{0,1\}^n} g(x) = \sum_{i=1}^{m} w_i x_i\,, \tag{7}$$

$$\text{s.t.} \sum_{j=1}^{m_s} \frac{\alpha_{j0}^s + \sum_{i=1}^n \alpha_{ji}^s x_i}{\beta_{j0}^s + \sum_{i=1}^n \beta_{ji}^s x_i} \ge p_s\,, \; s = 1, \dots, K\,, \tag{8}$$

where K is the number of fractional constraints.

Finally, we should note here that in contrast to (1)–(2), (6) and (7)–(8), problem (5) received most of the attention in the literature. Detailed surveys on single-ratio fractional combinatorial optimization can be found in [14,19].

Applications

Applications of constrained and unconstrained versions of problems (1)–(2), (5), (6), (7)–(8) arise in scheduling [16], query optimization in data bases and information retrieval [7], service systems design and facility location [3,20], graph theory [11], data mining [2] and other areas [19].

Consider, for example, a problem discussed in [3]. We have a set of customers' regions with Poisson demand rates $a_i (i = 1, \dots, n)$. These regions can be assigned to a service facility with an exponential service rate b. If we define a 0-1 variable x_i corresponding to each region i such that $x_i = 1$ if region i is serviced by the service facility (and $x_i = 0$, otherwise) then the service facility can be described as an $M/M/1$ queue with arrival rate $\lambda = \sum_{i=1}^n a_i x_i$ and service rate b. If we assume steady-state conditions ($\lambda < b$) then the average waiting time for each customer is equal to

$$\frac{1}{b - \lambda} = \frac{1}{b - \sum_{i=1}^n a_i x_i}\,, \tag{9}$$

and the total average waiting time is given by

$$\frac{\sum_{i=1}^n a_i x_i}{b - \sum_{i=1}^n a_i x_i}\,. \tag{10}$$

Next suppose that the customers' region i contributes profit p_i and the penalty for delay per unit time/per customer is t. Then in order to maximize the profit we need to solve the following nonlinear knapsack problem

$$\max_{x \in \{0,1\}^n} \sum_{i=1}^n p_i x_i - t \cdot \frac{\sum_{i=1}^n a_i x_i}{b - \sum_{i=1}^n a_i x_i}\,, \tag{11}$$

$$\text{s.t.} \sum_{i=1}^n a_i x_i \le b\,. \tag{12}$$

Another interesting application of fractional 0-1 programming can be found in graph theory [11]. Let $G = (V, E)$ be an undirected graph. The *density* $d(G)$ of G is defined as the maximum ratio of the number of edges e_H to the number of nodes n_H over all subgraphs $H \subseteq G$, i. e.

$$d(G) = \max_{H \subseteq G} \frac{e_H}{n_H}\,, \tag{13}$$

where e_H and n_H are the number of edges and nodes in the subgraph H. Obviously, the problem of finding $d(G)$ can be formulated as the following fractional 0-1 programming problem:

$$d(G) = \max_{x \in \{0,1\}^n,\; x \neq 0} \frac{\sum_{i=1}^n \sum_{j=1}^n a_{ij} x_i x_j}{2 \sum_{j=1}^n x_j}\,, \tag{14}$$

where a_{ij} are the elements of the adjacency matrix of G and n is the number of nodes in G. A similar formulation can also be given for the *arboricity* $\Gamma(G)$ which is defined as the minimum number of edge-disjoint forests into which G can be decomposed [11].

Complexity Issues

Constrained problems (1) and (5) where we optimize a single- or multiple-ratio fractional 0-1 function subject to linear 0-1 constraints, as well as problem (7)–(8) are obviously NP-hard since general linear 0-1 programming is their special case if we set $b_{ji} = 0$ and $b_{j0} = 1$ for $j = 1, \dots, m$ and $i = 1, \dots, n$.

An unconstrained single-ratio fractional 0-1 programming problem (5), can be solved in polynomial time, see [7], if condition (4) holds. If the denominator can take both negative and positive values, i. e., only (3) holds, single-ratio problem (5) is known to be NP-hard [7]. In other words, the sign of the denominator is "the borderline between polynomial and NP-hard classes" [7]. Another simple proof of this fact is given in [1]. Recall the classical **SUBSET SUM** problem: Given a set of positive integers $S = \{s_1, \dots, s_n\}$ and a positive integer K, does there exist a vector $x \in \{0,1\}^n$, such that

$$\sum_{i=1}^n s_i x_i = K\,? \tag{15}$$

This problem is known to be *NP*-complete [4]. With the instance of the **SUBSET SUM** problem we associate the following unconstrained single-ratio fractional 0-1 programming problem:

$$\max_{x\in\{0,1\}^n} \frac{1}{1-2(\sum_{i=1}^n s_i x_i - K)} . \tag{16}$$

It is easy to observe that (3) holds and the solution of (16) is equal to 1 if and only if the **SUBSET SUM** has a solution, which implies the necessary result. Furthermore, it can be easily shown that finding an approximate solution of (5) within any positive multiple of the (negative) optimal value is *NP*-hard [7].

For multiple-ratio problem (1) with (4) satisfied, the number of ratios ($m = 1$, or $m \geq 2$) defines complexity of the problem. For $m = 1$ we have a classical polynomially solvable single-ratio case, while for $m = 2$, that is the 2-ratio case, the problem becomes *NP*-hard (see [18] or [13]).

Some other aspects of the complexity of unconstrained single- and multiple-ratio fractional 0-1 programming problems (1) and (5), including complexity of uniqueness, approximability and local search, are addressed in [12,13].

Mixed Integer Reformulation

Li [9] and Wu [21] suggested a straightforward linearization technique for (1) based on a simple well-known idea: a polynomial mixed 0-1 term $z = xy$, where x is a 0-1 variable, and y is a continuous variable taking any positive value, can be represented by the following linear inequalities: (1) $y - z \leq K - Kx$; (2) $z \leq y$; (3) $z \leq Kx$; (4) $z \geq 0$, where K is an upper bound on y.

Assume that (4) is satisfied. Define a new variable y_j for each ratio in (1) that is

$$y_j = \frac{1}{b_{j0} + \sum_{i=1}^n b_{ji} x_i} . \tag{17}$$

Then fractional 0-1 programming problem (1) can be equivalently expressed as:

$$\begin{aligned}
&\max_{x\in\{0,1\}^n} \sum_{j=1}^m a_{j0} y_j + \sum_{j=1}^m \sum_{i=1}^n a_{ji} u_{ji} \\
&\text{s.t. } Dx \leq c \\
&b_{j0} y_j + \sum_{i=1}^n b_{ji} u_{ji} = 1 \quad j = 1, \ldots, m \\
&y_j - K_j(1 - x_i) \leq u_{ji} \leq K_j x_i \\
&\qquad j = 1, \ldots, m; i = 1, \ldots, n \\
&0 \leq u_{ji} \leq y_j \quad j = 1, \ldots, m;\ i = 1, \ldots, n ,
\end{aligned} \tag{18}$$

where a new variable u_{ij} is introduced for each nonlinear term $y_j x_i$, and K_j is an upper bound on y_j.

Additional, though similar in spirit to (18), linear mixed 0-1 reformulations as well as other related issues are carefully discussed in [20].

Solution Techniques

Most of the research efforts have been focused on solving various classes of single-ratio problem (5). Among developed solution techniques we should mention branch-and-bound [15], cutting plane [5], enumeration [6] and approximation algorithms [8]. However, most popular methods for solving single-ratio fractional 0-1 programming (and general fractional combinatorial) problems are based on the *parametric approach* [10,11,14].

For some classes of multiple-ratio fractional 0-1 programming problems, there are developed specialized algorithms [2,3,16,17,20]. More recent examples include a highly efficient cutting-plane algorithm for solving problem (11)–(12) [3] and a heuristic for solving special classes of fractionally constrained problems of type (7)–(8) [2]. Reported computational experiments involved test instances with the size of up to 10,000 variables.

Unfortunately, the fractional programming problem becomes substantially more difficult if we introduce additional ratios in the objective function. General multiple-ratio problem (1)–(2) can be solved utilizing standard branch-and-bound methods after reformulation into linear mixed 0-1 programming problem via techniques discussed in [9,20,21]. An improved branch-and-bound algorithm based on *node tightening* is developed in [20].

References

1. Boros E, Hammer P (2002) Pseudo-Boolean Optimization. Discret Appl Math 123(1-3):155–225

2. Busygin S, Prokopyev OA, Pardalos PM (2005) Feature Selection for Consistent Biclustering via Fractional 0-1 Programming. J Comb Optim 10(1):7–21
3. Elhedhli S (2005) Exact solution of a class of nonlinear knapsack problems. Oper Res Lett 33:615–624
4. Garey MR, Johnson DS (1979) Computers and Intractability: A Guide to the Theory of *NP*-Completeness. WH Freeman, San Francisco
5. Granot D, Granot F (1976) On solving fractional (0-1) programs by implicit enumeration. INFOR 14:241–249
6. Granot D, Granot F (1977) On integer and mixed integer fractional programming problems. Ann Discret Math 1:221–231
7. Hansen P, Poggi de Aragão M, Ribeiro CC (1991) Hyperbolic 0-1 programming and query optimization in information retrieval. Math Program 52:256–263
8. Hashizume S, Fukushima M, Katoh N, Ibaraki T (1987) Approximation algorithms for combinatorial fractional programming problems. Math Program 37:255–267
9. Li H (1994) A global approach for general 0-1 fractional programming. Eur J Oper Res 73:590–596
10. Megiddo N (1979) Combinatorial optimization with rational objective functions. Math Oper Res 4:414–424
11. Picard J-C, Queyranne M (1982) A network flow solution to some nonlinear 0-1 programming problems, with applications to graph theory. Networks 12:141–159
12. Prokopyev OA, Huang HX, Pardalos PM (2005) On complexity of unconstrained hyperbolic 0-1 programming problems. Oper Res Lett 33:312–318
13. Prokopyev OA, Meneses C, Oliveira CAS, Pardalos PM (2005) On Multiple-ratio Hyperbolic 0-1 Programming Problems. Pac J Optim 1/2:327–345
14. Radzik T (1998) Fractional Combinatorial Optimization. In: Du D-Z, Pardalos PM (eds) Handbook of Combinatorial Optimization, vol 1. Kluwer, Dordrecht, pp 429–478
15. Robillard P (1971) (0,1) Hyperbolic programming problems. Nav Res Logist Q 18:47–57
16. Saipe S (1975) Solving a (0,1) hyperbolic program by branch and bound. Nav Res Logist Q 22:497–515
17. Skiscim CC, Palocsay SW (2001) Minimum Spanning Trees with Sums of Ratios. J Glob Optim 19:103–120
18. Skiscim CC, Palocsay SW (2004) The Complexity of Minimum Ratio Spanning Tree Problems. J Glob Optim 30:335–346
19. Stancu-Minasian IM (1987) Fractional Programming. Kluwer, Dordrecht
20. Tawarmalani M, Ahmed S, Sahinidis N (2002) Global Optimization of 0-1 Hyperbolic Programs. J Glob Optim 24:385–416
21. Wu T-H (1997) A note on a global approach for general 0-1 fractional programming. Eur J Oper Res 101:1997

Frank–Wolfe Algorithm

SIRIPHONG LAWPHONGPANICH
Naval Postgraduate School, Monterey, USA

MSC2000: 90C30

Article Outline

Keywords
See also
References

Keywords

Away direction; Bisection search; Bounded; Column generation; Convergence rate; Convex function; Convex hull; Convexity; Dantzig–Wolfe decomposition; Differentiable function; Direction finding problem; Extreme point; Feasible direction; Feasible direction methods; First order Taylor series expansion; Geometrically; Globally optimal; Golden section method; Heuristics; Inexact line search line; Search algorithms; Line search problem; Linear program; Matrix; Network; Nonempty; Nonlinear programming; Parallel tangents; PARTAN; Polyhedron; Positive definite matrix; Pseudoconvex; Quadratic programming; Regularized direction finding problem; Regularized Frank–Wolfe decomposition; Simplex algorithm; Simplicial decomposition; Stepsize; Stopping criterion; Strongly convex

In 1956, M. Frank and P. Wolfe [5] published an article proposing an algorithm for solving quadratic programming problems. In the same article, they extended their algorithm to the following problem:

$$\min_{x \in S} f(x), \tag{1}$$

where $f(x)$ is a convex and continuously differentiable function on $\mathbf{R}^n$. The set S is a nonempty and bounded polyhedron of the form $\mathrm{S} = \{\mathrm{x} \in \mathbf{R}^n : \mathrm{Ax} \leq b, x \geq 0\}$, where A is a $m \times n$ matrix and $b \in \mathbf{R}^m$. The algorithm belongs to the class of *feasible direction methods for nonlinear programming problems*. Starting from a feasible solution, algorithms in this class solve (1) by iteratively generating a feasible direction that leads to another feasible solution with an improved objective function value. The Frank–Wolfe (FW) algorithm for (1) can be stated as follows:

```
0   Select $x^1 \in S$ and set $k = 1$.
1   Let $y_k = \arg\min_{y \in S} \nabla f(x^k)^\top y$.
    IF $\nabla f(x^k)^\top (y^k - x^k) \geq 0$
    THEN stop and $x^k$ is an optimal solution
    ELSE go to Step 2.
2   Let $\lambda^k = \arg\min_{0 \leq \lambda \leq 1} f(x^k + \lambda(y^k - x^k))$.
    Set $x^{k+1} = x^k + \lambda^k(y^k - x^k)$ and $k = k + 1$;
    Go to Step 1.
```

The Frank–Wolfe algorithm

The problem in Step 1 is generally referred to as the *direction finding problem*, for the vector $d^k = (y^k - x^k)$ is a feasible direction at x^k. Since $\nabla f(x^k)$ is a constant vector with respect to y, the direction finding problem is a linear program and can be solved using the simplex algorithm. Doing so implies that d^k always points toward an extreme point since y^k is always an extreme point of S. When x^k satisfies the stopping criterion, it must be globally optimal because the following holds for all $x \in S$:

$$\begin{aligned} f(x) &\geq f(x^k) + \nabla f(x^k)^\top (x - x^k) \\ &\geq f(x^k) + \nabla f(x^k)^\top (y^k - x^k) \geq f(x^k). \end{aligned}$$

The three inequalities follow from the convexity of $f(x)$, the fact that y^k solves the direction finding problem, and the stopping criterion, respectively.

When x^k does not satisfy the stopping criterion, $\nabla f(x^k)^\intercal(y^k - x^k) < 0$ and the algorithm proceeds to Step 2. In this step, λ^k is a solution to a line search problem which has only one a decision variable and can be solved by a number of algorithms such as bisection search, golden section method and an inexact line search technique using, e. g., Armijo's rule [1]. It is important to note that the new solution, x^{k+1}, has a better objective value. To demonstrate, consider the first order Taylor series expansion of $f(x)$ around the point x^k, i. e.,

$$\begin{aligned} & f(x^k + \lambda(y^k - x^k)) \\ &= f(x^k) + \lambda \nabla f(x^k)^\top (y^k - x^k) \\ &\quad + \lambda \left\| y^k - x^k \right\| \alpha(x^k; \lambda(y^k - x^k)), \end{aligned}$$

where $\lim_{\lambda \to 0} \alpha(x^k; \lambda(y^k - x^k)) = 0$. Since $\nabla f(x^k)^\intercal(y^k - x^k) < 0$, the above expansion implies that there exists a sufficiently small $\widehat{\lambda} \in (0, 1)$ such that $f(x^k + \widehat{\lambda}(y^k - x^k)) < f(x^k)$. Using the fact that λ^k solves the line search problem, the following must hold:

$$\begin{aligned} f(x^{k+1}) &= f(x^k + \lambda^k(y^k - x^k)) \\ &\leq f(x^k + \widehat{\lambda}(y^k - x^k)) < f(x^k). \end{aligned}$$

Thus, x^{k+1} has a better objective value.

Using standard techniques in nonlinear programming, it can be shown that the sequence of FW iterates, x^k, converges to an optimal solution. This also holds under a weaker assumption that $f(x)$ is pseudoconvex. In [14], W.B. Powell and Y. Sheffi eliminate the line search problem in Step 2 and show that the FW algorithm still converges to an optimal solution as long as λ^k satisfies the following conditions:

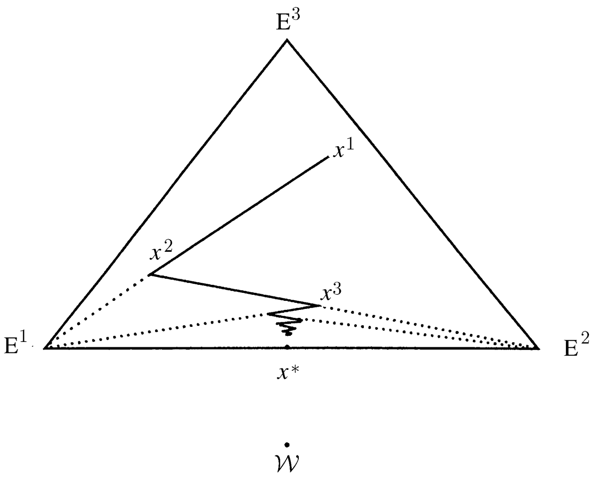
$$\sum_{k=1}^{\infty} \lambda^k \quad \text{and} \quad \lim_{k \to \infty} \lambda^k = 0.$$

For example, one suitable choice is $\lambda^k = 1/\mathrm{k}$.

The main advantage of the FW algorithm is in its simplicity. It is easy to understand and implement on a computer. Computer programs for the simplex and the line search algorithms already exist and are generally available. When the constraint matrix A has a network structure (see, e. g., [7,11], and [2]), more efficient network algorithms can be used to solve the direction finding problem and the overall computational time can be reduced. In addition, the FW algorithm does not require much computer storage or memory. However, this feature may be less important as the computer memory becomes available in abundance and at a cheaper price.

The main disadvantage of the FW algorithm is its slow convergence rate. (See Fig. 1.) During the early iterations, the algorithm tends to decrease the objective function rather dramatically. However, the FW iterates tend to zigzag as they slowly approach an optimal solution. In [17], Wolfe shows that the sequence x^k converges geometrically to the optimal solution, if it is in the relative interior of S and $f(x)$ is strongly convex. On the other hand, if the optimal solution is on the boundary of S, the convergence may be slower.

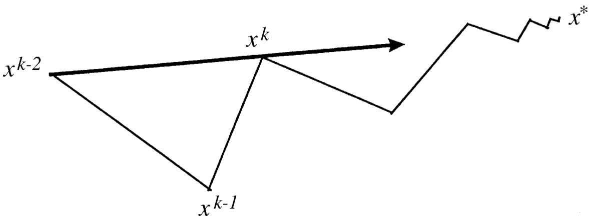
In practice, there are several modifications that can accelerate the convergence of the FW algorithm. The



Frank–Wolfe Algorithm, Figure 1
The problem is: min $\{\|w - x\|^2 : x \in S\}$ where S is the convex hull of E^1, E^2, and E^3. The Frank–Wolfe algorithm generates feasible directions that point toward either E^1 or E^2. It dramatically reduces the objective function during the first two iterations and zigzags toward the optimal solution, x^*, afterward

first modification is due to Wolfe [17]. It involves generating in Step 1 an additional feasible direction, $\widehat{d}^k = z^k - x^k$, where $z^k = \arg\max_{z \in S} \nabla f(x^k)_\top z$. The direction $\widehat{d}^k$ is generally referred to as the‘away’ direction since it is constructed from the worst extreme point with respect to minimizing the objective function. Between the original and the away directions, only one is selected for the line search problem in Step 2. Although the away direction generally leads to a faster convergence in practical applications (see, e. g., [3]), J. Guélat and P. Marcotte [8] showed that the resulting algorithm still converges geometrically to an optimal solution under appropriate assumptions. The second modification is based on the parallel tangents (PARTAN) method introduced in [15]. During the kth iteration, the PARTAN direction, p^k, is defined to be $(x^k - x^{k-2})$ when $k \geq 3$. When the FW algorithm zigzags, p^k intuitively points toward an optimal solution. (See Fig. 2.)

When integrated together, the PARTAN variant (see [4] and [10]) of the FW algorithm alternates between the original and the PARTAN directions when performing line searches. More formally, the original Step 2 of the FW algorithm is replaced with the following steps:



Frank–Wolfe Algorithm, Figure 2
The PARTAN direction, $p^k = (x^k - x^{k-2})$, points toward an optimal solution

```
2   Let $\lambda^k = \arg\min_{0 \leq \lambda \leq 1} f(x^k + \lambda(y^k - x^k))$.
    Set $z^k = x^k + \lambda^k(y^k - x^k)$.
    go to Step 3.
3   (PARTAN step)
    IF $k = 1$
    THEN set $x^{k+1} = z^k$
    ELSE let

        $\alpha^k = \arg\min_{0 \leq \alpha \leq \alpha^k_{\max}} f(x^{k-1} + \alpha(z^k - x^{k-1}))$,

        where $\alpha^k_{\max}$ is the maximal stepsize in the
        direction $(z^k - x^{k-1})$,
        set $x^{k+1} = x^{k-1} + \alpha^k(z^k - x^{k-1})$;
        set $k = k + 1$;
        return to Step 1.
```

Finally, the last modification for accelerating the FW algorithm involves using some or all of the extreme points generated during the current and prior iterations. Instead of performing a line search in Step 2, a typical modification (see, e. g., [6,9,16] and [12]) either requires a heuristic, approximate, or exact solution to the following problem:

$$\begin{cases} \min & f\left(\sum_{i=1}^{k} \beta_i y^i\right) \\ \text{s.t.} & \sum_{i=1}^{k} \beta_i = 1, \\ & \beta_i \geq 0, \quad i = 1, \ldots, k. \end{cases} \tag{2}$$

The feasible region of (2) is the convex hull of $\{y^1, \ldots, y^k\}$, each of which is an extreme point of S. Thus, (2) is an approximation to (1) and this approximation improves as more extreme points are added to (2). Since the number of extreme points of S is finite, an optimal solution to (2) should also solve (1) after a finite number of iterations. When (2) is solved exactly (or nearly so), the resulting algorithm is generally known as the *simplicial decomposition* or *column generation* technique and is related to the *Dantzig–Wolfe decomposition*.

In the above three modifications, the direction finding problems are linear programs with the same structure. In 1994, A. Migdalas [13] introduced an extension called the regularized Frank–Wolfe algorithm in which the direction finding problem has a nonlinear term in the objective function to control the distance between y^k and x^k. For example, one version of the regularized direction finding problem is:

$$y^k = \arg\min_{y \in S} \nabla f(x^k)^\top y + \frac{1}{2}(y - x^k)^\top D^k (y - x^k),$$

where D^k is a positive definite matrix.

See also

► Rosen's Method, Global Convergence, and Powell's Conjecture

References

1. Armijo L (1966) Minimization of functions having Lipschitz continuous first-partial derivatives. Pacific J Math 16:1–3
2. Collins M, Cooper L, Helgason R, Kennington J, LeBlanc L (1978) Solving the pipe network analysis problem using optimization techniques. Managem Sci 24:747–760
3. Florian M (1977) An improved linear approximation algorithm for the network equilibrium (packet switching) problem. Proc. 1977 IEEE Conf. on Decision and Control, New Orleans, Louisiana, pp 167–196
4. Florian M, Guéelat J, Spiess H (1987) An efficient implementation of the PARTAN variant of the linear approximation method of the network equilibrium problem. Networks 17:319–339
5. Frank M, Wolfe P (1956) An algorithm for quadratic programming. Naval Res Logist Quart 3:95–110
6. Fukushima M (1984) A modified Frank–Wolfe algorithm for solving the traffic assignment problem. Transport Res 18B:169–177
7. Golden BL (1975) A minimum-cost network flow problem concerning imports and exports. Networks 5:331–356
8. Guélat J, Marcotte P (1986) Some comments on Wolfe's away step. Math Program 35:110–119
9. Holloway C (1974) An extension of the Frank–Wolfe method of feasible directions. Math Program 6:14–27
10. LeBlanc LJ, Helgason RV, Boyce DE (1985) Improved efficiency of the Frank-Wolfe algorithm for convex network programs. Transport Sci 19:445–462
11. LeBlanc LJ, Morlok EK, Pierskalla WP (1975) An efficient approach to solving the road network equilibrium traffic assignment problem. Transport Res 9:308–318
12. Meyer G (1974) Accelerated Frank–Wolfe algorithms. SIAM J Control 12:655–663
13. Migdalas A (1994) A regularization of the Frank–Wolfe method and unification of certain nonlinear programming method. Math Program 65:331–346
14. Powell WB, Sheffi Y (1982) The convergence of equilibrium algorithms with predetermined step size. Transport Sci 16:45–55
15. Shah B, Buehler R, Kempthorne O (1964) Some algorithms for minimizing a function of several variables. SIAM J Appl Math 12:74–92
16. Von Hohenbalken B (1977) Simplicial decomposition in nonlinear programming algorithms. Math Program 13:49–68
17. Wolfe P (1970) Convergence theory in nonlinear programming. In: Abadie J (ed) Integer and Nonlinear Programming. North-Holland, Amsterdam, pp 1–36

Frequency Assignment Problem

FAP

ROBERT MURPHEY
US Air Force Research Labor., Eglin AFB, USA

MSC2000: 05-XX

Article Outline

Keywords
See also
References

Keywords

Optimization; Graph coloring

The ever growing number of wireless communications systems deployed around the globe has made the optimal assignment of a limited radio frequency spectrum a problem of primary importance. At issue are planning models for permanent spectrum allocation, licensing, regulation [20] and network design to include; aeronautical mobile, land mobile, maritime mobile, broadcast, land fixed (point-to-point) and satellite. Further at issue are on-line algorithms for dynamically assigning frequencies to users within an established network. In particular, land cellular mobile systems have been well studied (I. Katzela and M. Naghshineh [9] reference nearly 100 works in cellular dynamic channel assignment).

Frequency assignment problems are typically modeled in *graph theoretical* terms. That is, a graph $G(V, E)$ is considered with vertices $V(G) = \{v_1, \dots, v_n\}$ and edges $E(G)$. Each vertex in $V(G)$ represents a transmitter and two vertices (v_i, v_j) are adjacent (have an edge between them) if the corresponding transmitters are not permitted to share the same frequency. The frequency continuum is partitioned into channels (frequencies) of even width and numbered consecutive integer values. A frequency assignment is then a mapping f of the nonzero positive integers $\mathbf{Z}_+$ to the vertices of the graph such that no two adjacent vertices receive the same value:

$$\begin{aligned}&f\colon\ V \to \mathbb{Z}_+\\&\text{s.t. } (v_i, v_j) \in E(G) \Leftrightarrow f(v_i) \neq f(v_j).\end{aligned}$$

This formulation, where adjacent vertices cannot share the same frequency is termed *co-channel constrained* and was shown by B.H. Metzger [12] to be equivalent to the well-studied *graph coloring problem*. Typically, the objective is to find an assignment of frequencies (colors) to the transmitters (vertices) that minimizes the number of frequencies (colors) used. The minimum number $\chi(G)$ for which a $\chi(G)$-coloring exists for G is called the *chromatic number*. Since graph K-colorability for arbitrary K is known to be an *NP*-complete problem [6], co-channel constrained frequency assignment is also *NP*-complete.

Consider the restriction that two adjacent vertices may not receive frequencies that are the same or differ by exactly k. This FAP is said to be *adjacent channel constrained* and when $k = 0$ is simply the co-channel problem. Adjacent channel constraints model harmonic interference (signals that are integer multiples of the fundamental or carrier frequency). In general, a set T may be defined which contains zero and a subset of the positive integers such that no two adjacent vertices may receive assignments whose absolute difference is contained in T,

$$\begin{aligned}&f\colon\ V \to \mathbb{Z}_+\\&\text{s.t. } (v_i, v_j) \in E(G) \Leftrightarrow \left|f(v_i) - f(v_j)\right| \notin T.\end{aligned}$$

This FAP formulation was introduced by W.K. Hale [7] and is termed *T-coloring*. When $T = \{0\}$, the co-channel constrained FAP or graph coloring problem results. M.B. Cozzens and F.S. Roberts [4] define the number of unique colors used in a T-coloring as the *order* and the total bandwidth used (maximum color minus the minimum color) as the *span*. Hence for any T-coloring, two optimality criteria exist: minimum order, denoted by $\chi_T(G)$, and minimum span, denoted by $\mathrm{sp}_T(G)$. For the co-channel constrained FAP $\chi_T(G) = \mathrm{sp}_T(G)$ however, in general, this is not true. Cozzens and Roberts show that for any graph and any T the minimum order is equivalent to the chromatic number; $\chi_T(G) = \chi(G)$. Hence, T-coloring research has primarily been focused on characterizing the minimum span using numerous assumptions about the structure of G and value of T [2,4,5,11,13,16], and [17].

In many situations, the potential for interference between transmitters may occur on several different levels, where each level is defined by a separate set of edges on the common set of vertices. The kth edge set is denoted by the graph G_k, $k = 0, \dots, K$. The family of graphs thus defined and which share an identical vertex set are sometimes referred to, in unison, as a *multigraph* and denoted by $G(V, G_0, \dots, G_K)$. Since each level represents a unique interference mechanism, a family of T-sets must be also defined as $T(0), \dots, T(K)$. Interference occurs on the kth level when any 2 vertices adjacent in the kth edge set receive frequencies that differ by a value in $T(k)$. In graph coloring nomenclature, the multilevel FAP is denoted by

$$\begin{aligned}&f\colon\ V \to \mathbb{Z}_+,\\&(x, y) \in E(G_k) \Leftrightarrow |f(x) - f(y)| \notin T(k),\\&\forall (x, y) \in V, \quad x \neq y, \quad k = 0, \dots, K,\end{aligned}$$

where the family of graphs are nested such that $G_0 \supseteq \cdots \supseteq G_K$ and the T-sets are reverse nested, as $0 \in T(0) \subseteq \cdots \subseteq T(K)$. Cozzens and D.I. Wang develop bounds on the minimum span for general multigraph T-colorings in [5]. Excellent reviews on T-coloring and frequency assignment for single graphs and multigraphs may be found in [14] and [15].

Since the simplest FAP has been shown to be *NP*-complete, it is generally hopeless to pursue exact solution methods. Approximate heuristic techniques have been the focus of most research and most of these techniques fall under the scope of *sequential heuristics*. There are three fundamental approaches to sequentially coloring the vertices of a graph:

- (*frequency exhaustive*) Given an ordering of the vertices, attempt to color each vertex, sequentially, the smallest feasible color. This approach is also called a *greedy coloring*.
- (*requirement exhaustive*) Given an ordering of the vertices, attempt to assign the first color to each vertex, sequentially. When all vertices have been considered, attempt to assign the second color to the unassigned vertices, then the third and so on, following the same vertex ordering.
- (*uniform*) Given an ordering of the vertices, attempt to color each vertex, sequentially, the color that has been least used.

The efficiency of each approach is quite dependent upon what ordering the vertices are placed in. There are many rules by which the vertices of a graph can be ordered. In a smallest-last ordering, the vertex of smallest degree in V is denoted v_1. This vertex is then deleted from the graph and the next smallest degree vertex v_2 is found and deleted, and so on until all vertices have been deleted. The smallest-last vertex order is then $\{v_n, v_{n-1}, \ldots, v_1\}$. The largest-first vertex order sorts the vertices of the graph according to their degree in G: largest to smallest. D. Brelaz [3] introduced a vertex ordering specified by the saturation degree of the vertices, from highest saturation degree to lowest. The saturation degree of a vertex is defined to be the number of different colors that exist on the vertices that are adjacent. The vertex with the highest saturation degree is 'most denied' since it has fewer colors to choose from. J.A. Zoellner and C.L. Beall [21] compared the three sequential approaches with several different vertex ordering rules and found that, all else being equal, frequency exhaustive methods typically yields smaller spans. Hale [8] expanded upon these results by defining a generalized structure for all sequential coloring algorithms which consists of three fundamental steps:

1) order the vertices;
2) select the next vertex to color;
3) select the color.

Hale's procedure is general. It cover all types of vertex orderings in step 1 and allows for each of the three sequential techniques in step 3. Step 2 is added to allow the coloring sequence to adapt during the process. Hale introduced new sequential techniques for step 2 that are adaptive variants of the saturation degree. A very good review of frequency assignment heuristics can be found in [10].

Approximate solutions may also be obtained by using more traditional *polyhedral methods* on relaxation problems of the *integer program* (IP) formulation of the FAP. A. Wisse [19] developed a minimum order IP formulation for the FAP which relies on a *list coloring* model, that is, the frequencies which may be assigned are restricted to a finite list (set), designated by F, of cardinality m. Furthermore, I is designated as the index set for all transmitters (vertices) and n the cardinality of I. Define two binary decision variables as

$$x_{if} = \begin{cases} 1 & \text{if transmitter } i \text{ assigned freq } f, \\ 0 & \text{else,} \end{cases}$$

$$y_f = \begin{cases} 1 & \text{if freq } f \text{ used at least once,} \\ 0 & \text{else.} \end{cases}$$

The IP which results is

$$\begin{cases} \min & z = \sum_{f \in F} y_f \\ \text{s.t.} & \sum_{f \in F} x_{if} = 1, \quad \forall i \in I, \\ & \sum_{i \in I} x_{if} \le n y_f, \quad \forall f \in F, \\ & \sum_{g:\, |f-g| \notin T} x_{jg} \le 1 - x_{if}, \\ & \quad \forall f \in F, \quad \forall (v_i, v_j) \in E(G), \\ & x_{if} \in \{0,1\}, \quad \forall i \in I,\ \forall f \in F \\ & y_f \in \{0,1\}, \quad \forall f \in F. \end{cases}$$

A minimum span FAP IP formulation may be had by deleting variable y_f and adding μ, defined to be the maximum frequency assigned,

$$\begin{cases} \min & \mu \\ \text{s.t.} & \sum_{f \in F} x_{if} = 1, \quad \forall i \in I, \\ & \sum_{g:\, |f-g| \notin T} x_{jg} \le 1 - x_{if}, \\ & \quad \forall f \in F, \quad \forall (v_i, v_j) \in E(G), \\ & \sum_{f \in F} f x_{if} \le \mu, \quad \forall i \in I, \\ & x_{if} \in \{0,1\}, \quad \forall i \in I, \quad \forall f \in F, \\ & \mu \in F. \end{cases}$$

Of course a solution to either IP formulation would be exact, however efficient methods for finding solutions to this formulation do not yet exist for problems of large dimension. Linear relaxations of these formulations where $0 \leq x_{if} \leq 1$, have been successfully developed and yield fairly good solutions for some moderate size problems [1]. A *potential reduction* method [18] has also been developed that utilizes the transformation $x_{if} = 2x_{if} - 1$, that is, $x_{if} \in \{-1, +1\}$. As a result, any feasible solution to the transformed IPs must satisfy $x^{\mathsf{T}}x = mn$ where x is a vector of x_{if} in $\mathbf{Z}_{+}^{mn}$. The polyhedron $\mathcal{P}$ formed from the linear relaxation of x and y or μ in the constraints of the IPs is then incorporated into the problem by minimizing a logarithmic potential function over the polyhedron as

$$\min_{\mathcal{P}} \left[nm - x^{\top}x - \frac{1}{N}\sum_{k=1}^{N} w_k \log s_k \right],$$

where N is the number of constraints, w_k are nonnegative real valued weights, and s_k is the slack of constraint k. A sequence of iterative solutions are obtained in a three step process which begins with a nonoptimal feasible solution x^0. An interior point method is applied to a quadratic approximation of the potential function within an ellipsoid centered on the current feasible point x^i. This yields a decent direction Δx. The potential function is then minimized within the ellipsoid along the line $x^i + \alpha\Delta x$ and yields the next iterate x^{i+1}. The iterate solution is then rounded to an integer value. The algorithm stops when the rounded solution is feasible to the original problem. This algorithm was tested and was found to suffer from slow convergence. As a result, an alternate *quadratic assignment* formulation was developed which proved to be much faster. Define a new binary valued decision variable

$$q_{ifjg} = \begin{cases} 1 & \text{if } x_{if} = 1,\ x_{jg} = 1, \text{ and} \\ & \quad (v_i, v_j) \in E(G), \quad |f - g| \in T, \\ 0 & \text{else.} \end{cases}$$

Then the assignment $F \rightarrow x$ has no interference if

$$\sum_{i=1}^{n}\sum_{f=1}^{m}\sum_{j=1}^{n}\sum_{g=1}^{m} x_{if}x_{jg}q_{ifjg} = 0,$$

which is equivalent to

$$\tfrac{1}{2}x^{\top}Qx = 0,$$

where Q is a $mn \times mn$ matrix containing q_{ifjg}. Thus the new potential function, with the added quadratic term, is minimized over the polyhedron as

$$\min_{\mathcal{P}} \left[\frac{1}{2}x^{\top}Qx - \frac{1}{N}\sum_{k=1}^{N} w_k \log s_k \right].$$

Interior point solutions of this potential function converged much more quickly than those of the first formulation.

See also

- ▶ Assignment and Matching
- ▶ Assignment Methods in Clustering
- ▶ Bi-objective Assignment Problem
- ▶ Communication Network Assignment Problem
- ▶ Graph Coloring
- ▶ Maximum Constraint Satisfaction: Relaxations and Upper Bounds
- ▶ Maximum Partition Matching
- ▶ Quadratic Assignment Problem

References

1. Aardal KI, Hipolito A, Van Hoesel CPM, Jansen B, Roos C, Terlaky T (1995) A branch-and-cut algorithm for the frequency assignment problem. Techn. Report EUCLID CALMA Project, Delft and Eindhoven Univ. Techn., The Netherlands
2. Bonias I (1991) T-colorings of complete graphs. PhD Thesis Dept. Math. Northeastern Univ.
3. Brelaz D (1979) New methods to color the vertices of a graph. Comm ACM 22:251–256
4. Cozzens MB, Roberts FS (1982) T-colorings of graphs and the channel assignment problem. Congressus Numerantium 35:191–208
5. Cozzens MB, Wang DI (1984) The general channel assignment problem. Congressus Numerantium 41:115–129
6. Garey MR, Johnson DS (1979) Computers and intractability. A guide to the theory of NP-completeness. Freeman, New York
7. Hale WK (1980) Frequency assignment: theory and applications. Proc IEEE 68(12):1497–1514
8. Hale WK (1981) New spectrum management tools. IEEE Internat. Symp. Electromagnetic Compatibility, pp 47–53
9. Katzela I, Naghshineh M (1996) Channel assignment schemes for cellular mobile telecommunication systems: A comprehensive survey. IEEE Personal Comm 33:10–31

10. Lanfear TA (1989) Graph theory and radio frequency assignment. Techn. Report Allied Radio Frequency Agency NATO
11. Liu DD (1991) Graph homomorphisms and the channel assignment problem. PhD Thesis Dept. Math. Univ. South Carolina
12. Metzger BH (1970) 38th Nat. ORSA Meeting. In: Spectrum management techniques
13. Raychaudhuri A (1985) Intersection assignments, T-coloring, and powers of graphs. PhD Thesis Dept. Math. Rutgers Univ.
14. Roberts FS (1991) From garbage to rainbows: generalizations of graph coloring and their applications. In: Alavi Y, Chartrand G, Oellerman OR, Schwenk AJ (eds) Graph Theory, Combinatories, and Applications, vol 2. Wiley, New York, pp 1031–1052
15. Roberts FS (1991) T-colorings of graphs: recent results and open problems. Discret Math 93:229–245
16. Tesman BA (1989) T-colorings, list T-colorings, and set T-colorings of graphs. PhD Thesis Dept. Math. Rutgers Univ.
17. Wang DI (1984) The channel assignment problem and closed neighborhood containment graphs. PhD Thesis Northeastern Univ.
18. Warners JP, Terlaky T, Roos C, Jansen B (1995) A potential reduction approach to the frequency assignment problem. Fac. Techn. Math. and Inform. Delft Univ., The Netherlands, 95–98
19. Wisse A (1994) Mathematical model for the radio link frequency assignment problem. EUCLID CALMA Project, Delft Univ. Techn., The Netherlands
20. Zoellner JA (1973) Frequency assignment games and strategies. IEEE Trans Electromagnetic Compatibility EMC-15:191–196
21. Zoellner JA, Beall CL (1977) A breakthrough in spectrum conserving frequency assignment technology. IEEE Trans Electromagnetic Compatibility EMC-19:313–319

Fundamental Theorem of Algebra

HELMUT MAIER
Abt. Math. III, University Ulm, Ulm, Germany

MSC2000: 01A55, 01A50, 01A60

Article Outline

Keywords
Analytic Proofs
Topological Proofs
Algebraic Proofs
See also
References

Keywords

Argument principle; Homotopy; Splitting field

Some of the main indicators of progress in the mathematical sciences have been the occurrences of new types of numbers. One of the more recent cases are the complex numbers. Much of modern science cannot be imagined without their use.

Their introduction into mathematics first had been motivated by the wish to solve the equation

$$P(z) = 0, \tag{1}$$

where $P(z)$ is a polynomial.

If one considers only real numbers, such simple equations like $P(z) = z^2 + 1 = 0$ have no solutions. In the field of complex numbers however (1) always at least one solution, if P is a nonconstant polynomial with complex coefficients. This fact is known as the fundamental theorem of algebra. It was first proved rigorously by C.F. Gauss in 1799. Since then a large number of proofs have been found. In this article I give some examples for the main types of proofs: analytic, topological and algebraic.

Analytic Proofs

Possibly the simplest proof, being based on the *Liouville theorem* [3]: Every bounded entire function is a constant.

Assume now that the nonconstant polynomial $P(z)$ has no zero. Since $|P(z)| \to \infty$ for $|z| \to \infty$, the function $f(z) = 1/P(z)$ is bounded and thus a constant by Liouville's theorem. But then also $P(z)$ is constant, a contradiction.

Another, still simple, proof is based on the *argument principle*: The number of zeros of a holomorphic function f inside a simple closed curve γ can be expressed by the integral

$$\frac{1}{2\pi i}\int_{\gamma}\frac{f'(z)}{f(z)}\,dz.$$

Let $P(z)$ be a polynomial of degree $n \geq 1$. Choosing for γ the circle around the origin with radius $R > 0$, we obtain for the number N of zeros of $P(z)$:

$$N = \frac{1}{2\pi i}\int_{\gamma}\frac{P'(z)}{P(z)}\,dz\,.$$

Since

$$\frac{P'(z)}{P(z)} = \frac{n}{z} + O\left(\frac{1}{|z|^2}\right), \quad |z| \to \infty,$$

we obtain $N = n$ for $R \geq \mathrm{R}_0$. Thus $P(z)$ has n zeros (counted with multiplicity).

Topological Proofs

Closely related to the second analytic proof, presented above, is the proof by the concept of homotopy [2].

If X and Y are two topological spaces, then two continuous maps $\varphi_0, \varphi_1 \colon X \to Y$ are called *homotopic* if there exists a continuous map

$$\varphi\colon\ X \times [0,1] \to Y$$

such that

$$\varphi(x,0) = \varphi_0(x),$$
$$\varphi(x,1) = \varphi_1(x).$$

We choose $X = \{z \in \mathrm{C}\colon |\mathrm{z}| = 1\}$, $\mathrm{Y} = \mathrm{C} - 0$. Let $P(z) = z^n + a_{n-1}z^{n-1} + \cdots + a_0 = z^n + Q(z)$, say $n \geq 1$, $a_0 \neq 0$).

For sufficiently large R the two maps

$$\varphi_0\colon\ X \to Y, \quad z \longmapsto (Rz)^n,$$
$$\varphi_1\colon\ X \to Y, \quad z \longmapsto P(Rz)$$

are homotopic. A homotopy is in fact given by $\varphi(z,t) = (Rz)^n + \mathrm{t}\,Q(Rz)$, $z \in X$, $t \in [0,1]$. If there is no zero of $P(z)$ inside the circle $|\mathrm{z}| = R$, φ_1 and thus also φ_0 would be homotopic with the constant map

$$\varphi_c\colon\ X \to Y, \quad z \longmapsto a_0;$$

which can be shown to be false by topological means.

Algebraic Proofs

Since there is no purely algebraic system of axioms for the field of complex numbers there cannot be a purely algebraic proof. However there is a proof which as the only result from analysis uses the intermediate value theorem [1], which we reproduce here.

A statement equivalent to the fundamental theorem is that **C** is the algebraic closure of **R**. We start by showing that every nonconstant polynomial $P(z)$ with *real coefficients* has a complex zero. We proceed by induction.

Let n be the degree of $P(z)$.

i) If n is odd, the claim is an immediate consequence of the intermediate value theorem.

ii) Let $n = 2^t u$ with odd u, $t > 0$, and assume the claim has been proven for $t - 1$.
We select a splitting field S for $P(z)$ over **C**. Then we have a decomposition

$$P(z) = (z - a_1)\cdots(z - a_n) \quad \text{in } S[z].$$

For an arbitrary real number c we form the expressions $b_{ij}(c) = a_i a_j + \mathrm{c}(a_i + a_j)$ and the polynomial $Q(z) = \prod_{1 \leq i < j \leq n} (z - b_{ij}(c))$. The coefficients of $Q(z)$ are symmetric polynomials in $a_1, \ldots, a_n$ over **R** and thus real. The degree of $Q(z)$ is $n(n-1)/2 = 2^{\mathrm{t}-1}u(2^t u - 1) = 2^{\mathrm{t}-1}v$ for an odd number v. By the induction hypothesis $Q(z)$ has at least one zero in **C**. Thus $b_{ij}(c)$ is in **C** for a pair of subscripts (i, j) that may depend on c. If this construction is carried out for all natural numbers c with $1 \leq c \leq 1 + n(n-1)/2$ one finds c and c' belonging to the same pair of subscripts, i. e. there is a pair (i, j) with $b_{ij}(c) \in \mathbf{C}$ and $b_{ij}(c') \in \mathbf{C}$. If one solves the system of equations

$$b_{ij}(c) = a_i a_j + c(a_i + a_j),$$
$$b_{ij}(c') = a_i a_j + c'(a_i + a_j)$$

one obtains $a_i = a/2 \pm \sqrt{a^2 - 4b^2}/2 \in \mathbb{C}$. Thus $P(z)$ has a complex zero.

Let now $P(z) \in \mathbf{C}[z]$ be irreducible and t a zero of $P(z)$ in a splitting field of $P(z)$ over **C**. Then $P(z)$ is the irreducible polynomial of t over **C**. Since t is algebraic over **C** and **C** is algebraic over **R**, t is algebraic over **R**. We denote the irreducible polynomial of t over **R** by $U(z)$. Then $P(z)/U(z)$ in $\mathbf{C}[z]$.

$U(z)$ has at least one zero in **C**. Since **C** is normal over **R**, $U(z)$ splits into linear factors in $\mathbf{C}[z]$. Thus $P(z)$ is linear and $t \in \mathbf{C}$.

See also

► Gröbner Bases for Polynomial Equations

References

1. Körner O (1990) Algebra. Aula-Verlag, Wiesbaden
2. Massey WS (1967) Algebraic topology: An introduction. Springer, Berlin
3. Titchmarsh EC (1939) The theory of functions. Oxford Univ. Press, Oxford

Fuzzy Multi-objective Linear Programming

FMOLP

ROMAN SLOWINSKI
Institute Computing Sci., Poznań University Technol., Poznań, Poland

MSC2000: 90C70, 90C29

Article Outline

Keywords
Flexible Programming
MOLP with Fuzzy Coefficients
Flexible MOLP with Fuzzy Coefficients
Conclusions
See also
References

Keywords

Multi-objective linear programming under uncertainty; Fuzzy sets; Uncertainty modeling; Multicriteria decision making; Interactive procedures

Fuzzy multi-objective linear programming extends the linear programming model (LP) in two important aspects:

- multiple objective functions representing different points of view (criteria) used for evaluation of feasible solutions,
- *uncertainty* inherent to information used in the modeling and solving stage.

A general model of the FMOLP problem can be presented as the following system:

$$[\widetilde{\mathbf{c}}_1\mathbf{x},\dots,\widetilde{\mathbf{c}}_k\mathbf{x}] \to \widetilde{\min} \tag{1}$$

such that

$$\widetilde{\mathbf{a}}_i\mathbf{x}\,\widetilde{\leq}\,\widetilde{b}_i, \quad i=1,\dots,m, \tag{2}$$

$$\mathbf{x} \geq 0, \tag{3}$$

where $\widetilde{\mathbf{c}}_1 = [\widetilde{c}_{l1},\dots,\widetilde{c}_{ln}$ $(l = 1,\dots,k)$, $\mathbf{x} = [x_1,\dots,x_n]^\intercal$, $\widetilde{\mathbf{a}}_i = [\widetilde{a}_{i1},\dots,\widetilde{a}_{in}$ $(i = 1,\dots,m)$. The coefficients with the sign of wave are, in general, *fuzzy numbers*, i. e. convex continuous fuzzy subsets of the real line. The wave over min and relation $\leq$ 'fuzzifies' their meaning. Conditions (2) and (3) define a set of feasible solutions (decisions) X. An additional information completing (1) is a set of fuzzy aspiration levels on particular objectives, thought of as goals, denoted by $\widetilde{g}_1,\dots,\widetilde{g}_k$.

There are three important special cases of the above problem that gave birth to the following classes of problems:

- flexible programming;
- multi-objective linear programming (MOLP) with fuzzy coefficients;
- flexible MOLP with fuzzy coefficients.

In *flexible programming*, coefficients are crisp but there is a fuzzified relation $\widetilde{\leq}$ between objective functions and goals, and between left- and right-hand sides of the constraints. This means that the goals and constraints are fuzzy ('soft') and the key question is the degree of satisfaction. In *MOLP with fuzzy coefficients* all the coefficientsare, in general, fuzzy numbers and the key question is a representation of relation $\leq$ between fuzzy left- and right-hand sides of the constraints. Flexible MOLP with fuzzy coefficients concerns the most general form (1)–(3) and combines the two key questions of the previous problems.

The two first classes of FMOLP problems use different semantics of *fuzzy sets* while the third class combines the two semantics. In flexible programming, fuzzy sets are used to express *preferences* concerning satisfaction of flexible constraints and/or attainment of goals. This semantics is especially important for exploiting information in decision making. The gradedness introduced by fuzzy sets refines the simple binary distinction made by ordinary constraints. It also refines the crisp specification of goals and 'all-or-nothing' decisions. Constraint satisfaction algorithms, optimization techniques and multicriteria decision analysis are typically involving flexible requirements which can be represented by fuzzy relations.

In MOLP with fuzzy coefficients, the semantics of fuzzy sets is related to the representation of incomplete or vague states of information under the form of possibility distributions. This view of fuzzy sets enables representation of *imprecise* or *uncertain information* in mathematical models of decision problems considered in operations research. In models formulated in terms of mathematical programming, the imprecision and uncertainty of information (data) is taken into account through the use of fuzzy numbers or fuzzy intervals instead of crisp coefficients. It involves fuzzy arithmetic and other mathematical operations on fuzzy numbers that are defined with respect to the famous Zadeh's extension principle.

In flexible MOLP with fuzzy coefficients, the uncertainty and the preference semantics are encountered together. This is typical for decision analysis and operations research where, in order to deal with both uncertain data and flexible requirements, one can use a fuzzy set representation.

Below, we make a tutorial characterization of the three classes of problems and solution methods. For more detailed surveys see, e. g., [16,18,20,27, 30,32,36,37].

Flexible Programming

Flexible programming has been considered for the first time in [41] with respect to single-objective linear programming. It is based on a general Bellman–Zadeh principle [2] defining the concept of *fuzzy decision* as an intersection of *fuzzy goals* and *fuzzy constraints*. A fuzzy goal corresponding to objective $\mathbf{c}_l\mathbf{x}$ is defined as a fuzzy set in X; its membership function $\mu_l : X \to [0,1]$ characterizes the decision maker's aspiration of making $\mathbf{c}_l\mathbf{x}$ 'essentially smaller or equal to g_l'. A fuzzy constraint corresponding to $\mathbf{a}_i\mathbf{x} \tilde{\leq} b_i$ is also defined as a fuzzy set in X; its membership function $\mu_i \to [0,1]$ characterizes the degree of satisfaction of the ith constraint.

In order to define the membership function $\mu_i(\mathbf{x})$ for the ith fuzzy constraint, one has to know the tolerance margin $d_i \geq 0$ for the right-hand side b_i $(i = 1,\dots,m)$;

$$\mu_i(\mathbf{x}) = \begin{cases} 1 & \text{for } \mathbf{a}_i\mathbf{x} \leq b_i\,, \\ \text{strictly} & \text{decreasing from 1 to 0} \\ & \ \text{for } b_i < \mathbf{a}_i\mathbf{x} < b_i + d_i\,, \\ 0 & \text{for } \mathbf{a}_i\mathbf{x} \geq b_i + d_i\,. \end{cases} \tag{4}$$

Specifying a membership level α, $\alpha \in [0,1]$, in [41] the set of feasible solutions ofeach fuzzy constraint has been restricted to the crisp set

$$X_\alpha^i = \{\mathbf{x} \colon\ \mu_i(\mathbf{x}) \geq \alpha\}\,, \quad i = l,\dots,m\,.$$

Then, the set of feasible solutions of a flexible programming problem is $X_\alpha = \cap_{i=1}^m X_\alpha^i$. The single objective function is replaced by the fuzzy goal

$$\mu_G(\mathbf{x}) = \frac{\min_{\mathbf{x}\in X_0}\{\mathbf{cx}\}}{\mathbf{cx}}\,.$$

To get an optimal solution one has to determine the optimal pair $(\alpha^*, \mathbf{x}^*)$ such that

$$\min\{\alpha^*, \mu(\mathbf{x}^*)\} = \sup \min\nolimits_\propto \left(\alpha, \max_{\mathbf{x}\in X_\alpha}\{\mu_G(\mathbf{x})\}\right)\,. \tag{5}$$

If the optimal α^* was determined a priori, the problem(5) could be reduced to a crisp mathematical programming problem where the objective was to find $\mathbf{x}^*$ that maximizes $\mu_G(\mathbf{x})$ on the set X_α^*. In the general case an iterative algorithm is necessary when beginning with any $\alpha_1 \in [0,1]$, the values α_k and $\max_{\mathbf{x}\in X_{\infty k}}\{\mu_G(\mathbf{x})\}$ converge to the optimum step by step.

H.J. Zimmermann [46] has proposed a more integrative approach to flexible programming allowing consideration of multiple goals and constraints on a common ground. An aspiration level g_l and a tolerance margin $d_l \geq 0$ have to be assumed for the lth goal $(l = 1,\dots,k)$ when assessing the membership function $\mu_l(\mathbf{x})$ as:

$$\mu_l(\mathbf{x}) = \begin{cases} 1 & \text{for } \mathbf{c}_l\mathbf{x} \leq g_l\,, \\ \text{strictly} & \text{decreasing from 1 to 0} \\ & \ \text{for } g_l < \mathbf{c}_l\mathbf{x} < g_l + d_l\,, \\ 0 & \text{for } \mathbf{c}_l\mathbf{x} \geq g_l + d_l\,. \end{cases} \tag{6}$$

According to the Bellman–Zadeh principle, the set of fuzzy decisions is characterized by an aggregation of the component membership functions. If a conjunctive minimum operator were used for the aggregation, the membership function would be:

$$\mu_D(\mathbf{x}) = \min_{l,i}\{\mu_l(\mathbf{x}), \mu_i(\mathbf{x})\}\,. \tag{7}$$

Then, the problem of finding the best decision (solution) boils down to the following optimization problem:

$$\begin{cases} \mu_D(\mathbf{x}) & \to \max \\ \text{s.t.} & \mathbf{x} \geq 0\,. \end{cases} \tag{8}$$

The value of the aggregated function $\mu_D(\mathbf{x})$ can be interpreted as the overall degree of satisfaction of the decision maker with k fuzzy goals and m fuzzy constraints.

In case of minimum operator (7), problem (8) becomes:

$$\begin{cases} v & \to \max \\ \text{s.t.} & v \leq \mu_l(\mathbf{x}), \quad l = 1, \ldots, k, \\ & v \leq \mu_i(\mathbf{x}), \quad i = 1, \ldots, m, \\ & \mathbf{x} \geq 0. \end{cases} \tag{9}$$

In [46,47], Zimmermann has applied linear membership functions (4), (6) in problem (9) thus getting an ordinary LP problem. He also proposed to use the product operator instead of minimum, however, then (8) becomes nonlinear even if linear membership functions are used. A comprehensive review of various propositions for modeling the functions $\mu_D(\mathbf{x})$ can be found in [39,48].

Knowing the membership functions $\mu_l(\mathbf{x})(l = 1,\ldots, k)$ for fuzzy goals, one can define a Pareto optimal solution in the space of membership values, calledan M-Pareto optimal solution [32]. Some other refinements of the Zimmermann's approach have been proposed in [1,11].

Definition 1 A solution $\mathbf{x}^*$ is said to be *M-Pareto optimal* if and only if there does not exist another $\mathbf{x} \in X$ such that $\mu_l(\mathbf{x}) \geq \mu_l(\mathbf{x}^*), l = 1, \ldots, k$, with strict inequality holding for at least one *l*.

The concept of M-Pareto optimal solutions was at the origin of several *interactive methods* proposed for flexible programming (see [30,32]). In these methods, the decision maker determines membership functions for fuzzy goals and then specifies reference levels for the membership functions, denoted by $\overline{\mu}_l$ $(l = 1, \ldots, k)$. Assuming some minimum levels for membershipfunctions of fuzzy constraints, denoted by t_i $(i = 1, \ldots, m)$, one gets the following optimization problem:

$$\begin{cases} \max\limits_l\{\overline{\mu}_l - \mu_l(\mathbf{x})\} & \to \min \\ \text{s.t.} & \mu_i(\mathbf{x}) \geq t_i, \ i = 1, \ldots, m, \\ & \mathbf{x} \geq 0, \end{cases}$$

which is equivalent to

$$\begin{cases} v & \to \min \\ \text{s.t.} & v \geq \overline{\mu}_l - \mu_l(\mathbf{x}), \quad l = 1, \ldots, k, \\ & \mu_i(\mathbf{x}) \geq t_i, \quad i = 1, \ldots, m, \\ & \mathbf{x} \geq 0. \end{cases} \tag{10}$$

Again, problem (10) becomes an ordinary LP problem when all membership functions are linear. This approach is interactive in the sense that the reference levels can be changed from one iteration to another, as well as the membership functionsof fuzzy goals.

MOLP with Fuzzy Coefficients

All fuzzy coefficients of the FMOLP problem are given in a convenient form of *L*-*R* fuzzy numbers [13]. An *L*-*R* *(flat) fuzzy number* $\widetilde{a} = (a^L, a^R, \alpha^L, \alpha^R)_{LR}$ is defined by the membership function:

$$\mu_{\tilde{a}}(r) = \begin{cases} L\left(\frac{a^L - r}{\alpha^L}\right) & \text{for } r \leq a^L, \\ 1 & \text{for } a^L \leq r \leq a^R, \\ R\left(\frac{r - a^R}{\alpha^R}\right) & \text{for } r \geq a^R, \end{cases}$$

where *L* and *R* are symmetric bell-shaped reference functions which are strictly decreasing in [0, 1] and such that $L(0) = \mathrm{R}(0) = 1$, $L(1) = \mathrm{R}(1) = 0$; $[a^L, a^R]$ is an interval of the most possible values, and α^L and α^R are nonnegative left and right 'spreads' of $\widetilde{a}$, respectively.

Experience indicates that an expert can describe the precise form of a fuzzy number only rarely. Therefore, as a practical way of getting suitable membership functions of fuzzy coefficients, H. Rommelfanger [26] has proposed that the expert begins with the specification of some prominent membership levels α and associates them with special meanings. After that the expert is expected tospecify values which belong to the selected membership levels.

α = 1: $\mu_{\tilde{a}}(r) = 1$ means that value *r* certainly belongs to the set of possible values;

α = λ: $\mu_{\tilde{a}}(r) \geq \lambda$ means that the expert estimates that value *r* with $\mu_{\tilde{a}}(r) \geq \lambda$ has a good chance of belonging to the set of possible values;

α = ε: $\mu_{\tilde{a}}(r) < \varepsilon$ means that value *r* with $\mu_{\tilde{a}}(r) < \varepsilon$ has only a very little chance of belonging to the set of possible values, i. e. the expert is willing to neglect the corresponding values of *r* with $\mu_{\tilde{a}}(r) < \varepsilon$.

For example, it is reasonable to assume that $\lambda = 0.6$, $\varepsilon = 0.1$.

For the sake of clarity, let us assume that the reference functions of all fuzzy coefficients are of two kinds only: *L* and *R*. It should be specified, moreover, that all *arithmetic operations on fuzzy numbers* taking place in (1), (2) are extended operations in the sense of Zadeh's extension principle [45]:

$$f_{\tilde{a}*\tilde{b}}(r) = \sup_{r=y*z} T(f_{\tilde{a}}(y), f_{\tilde{b}}(z)), \quad r \in \mathbb{R}, \tag{11}$$

where $*$ is a real operation $*$: $\mathbf{R} \times \mathbf{R} \to \mathbf{R}$ and T: $[0, 1] \times [0, 1] \to [0, 1]$ is any given t-norm.

For any $\mathbf{x} \geq 0$, the left-hand side of the ith constraint and the value of the lth objective function can be summarized to the following fuzzy numbers:

$$\begin{aligned} \widetilde{\mathbf{a}}_i\mathbf{x} &= \left(\mathbf{a}_i^L\mathbf{x}, \mathbf{a}_i^R\mathbf{x}, \alpha_i^L\mathbf{x}, \alpha_i^R\mathbf{x}\right)_{LR}, \quad & i = 1, \ldots, m, \\ \widetilde{\mathbf{c}}_l\mathbf{x} &= \left(\mathbf{c}_l^L\mathbf{x}, \mathbf{c}_l^R\mathbf{x}, \gamma_l^L\mathbf{x}, \gamma_l^R\mathbf{x}\right)_{LR}, \quad & l = 1, \ldots, k. \end{aligned}$$

In the literature the min t-norm is generally applied. Then,

$$\mathbf{a}_i^L\mathbf{x} = \sum_{j=1}^n a_{ij}^L x_j, \quad \mathbf{c}_l^L\mathbf{x} = \sum_{j=1}^n c_{lj}^L x_j, \tag{12}$$

$$\mathbf{a}_i^R\mathbf{x} = \sum_{j=1}^n a_{ij}^R x_j, \quad \mathbf{c}_l^R\mathbf{x} = \sum_{j=1}^n c_{lj}^R x_j, \tag{13}$$

$$\alpha_i^L\mathbf{x} = \sum_{j=1}^n \alpha_{ij}^L x_j, \quad \gamma_l^L\mathbf{x} = \sum_{j=1}^n \gamma_{lj}^L x_j, \tag{14}$$

$$\alpha_i^R\mathbf{x} = \sum_{j=1}^n \alpha_{ij}^R x_j, \quad \gamma_l^R\mathbf{x} = \sum_{j=1}^n \gamma_{lj}^R x_j. \tag{15}$$

Obviously, the spreads of these fuzzy numbers extend when number and values of variables increase. The simple addition of the spreads of fuzzy coefficients corresponds to the assumption that their uncertainty comes fromindependent sources. This is not realistic in many practical situations. For getting a more realistic extended addition of the left-hand sides of fuzzy constraints and of fuzzy objectives, Rommelfanger and T. Keresztfalvi [29] recommend the use of Yager's parameterized t-norm:

$$\begin{aligned} &T_p(t_1, \ldots, t_s) \\ &= \max\left\{0, 1 - \left(\sum_{i=1}^s (1-t_i)^p\right)^{1/p}\right\}, \\ &\qquad t_1, \ldots, t_s \in [0, 1], \quad p > 0. \end{aligned} \tag{16}$$

Then, $\mathbf{a}_i^L\mathbf{x}$, $\mathbf{a}_i^R\mathbf{x}$, $\mathbf{c}_l^L\mathbf{x}$, $\mathbf{c}_l^R\mathbf{x}$ are calculated according to (12) and (13), however, the spreads $\alpha_i^L\ \mathbf{x}$, $\alpha_i^R\mathbf{x}$, $\gamma_l^L\mathbf{x}$, $\gamma_l^R\ \mathbf{x}$ are calculated according to a new, less cumulative formula:

$$\alpha_i^L\mathbf{x} = \left(\sum_{j=1}^n \left(\alpha_{ij}^L x_j\right)^q\right)^{1/q},$$

$$\alpha_i^R\mathbf{x} = \left(\sum_{j=1}^n \left(\alpha_{ij}^R x_j\right)^q\right)^{1/q},$$

$$\gamma_i^L\mathbf{x} = \left(\sum_{j=1}^n \left(\gamma_{ij}^L x_j\right)^q\right)^{1/q},$$

$$\gamma_i^R\mathbf{x} = \left(\sum_{j=1}^n \left(\gamma_{ij}^R x_j\right)^q\right)^{1/q},$$

where $q = p/(p-1) \geq 1$.

Coming back to MOLP problem with fuzzy coefficients, we haveto answer the question how to interpret the relation between fuzzy left- and right-hand side of the constraints. If constraints (2) were transformed to equality constraints (by addition of slack variables on the left) thenthe equality relation could be interpreted in terms of weak inclusion of fuzzy sets [12,21]:

$$\widetilde{\mathbf{a}}_i\mathbf{x} \subseteq \widetilde{b}_i, \quad i = 1, \ldots, m. \tag{17}$$

It says that the region of possible values of the left-hand side should be contained in the tolerance region of the right-hand side. The LP problem with constraints (17) is called *robust programming problem.*

Each constraint (18) is then reduced to four deterministic constraints:

$$\begin{aligned} &\mathbf{a}_i^L\mathbf{x} \geq b_i^L, \quad \mathbf{a}_i^R\mathbf{x} \leq b_i^R, \\ &\mathbf{a}_i^L\mathbf{x} - \alpha_i^L\mathbf{x} \geq b_i^L - \beta_i^L, \\ &\mathbf{a}_i^R\mathbf{x} + \alpha_i^L\mathbf{x} \leq b_i^R + \beta_i^R, \\ &\text{for } i = 1, \ldots, m, \end{aligned} \tag{18}$$

where $\widetilde{b}_i = (b_i^L, b_i^R, \beta_i^L, \beta_i^R)_{LL}$ or $\widetilde{b}_i = (b_i^L, b_i^R, \beta_i^L, \beta_i^R)_{RR}$, $i = 1, \ldots, m$.

In order to transform fuzzy objectives into deterministic equivalents, one can consider a 'middle' value of $\widetilde{\mathbf{c}}_l\mathbf{x}$ at some level $\xi \in [0, 1]$, $l = 1, \ldots, k$. The 'middle' can be understood [8] as a weighted combination of the most possible values $\mathbf{c}_l^L\mathbf{x}$ and $\mathbf{c}_l^R\mathbf{x}$, and of the smallest and the greatest (extreme) values at possibility level ξ. Thus, the objectives (1) become:

$$[z_1(\mathbf{x}), \ldots, z_k(\mathbf{x})] \to \min, \tag{19}$$

where $z_l(\mathbf{x}) = w_1\mathbf{c}_l^L - w_2\gamma_l^L\mathbf{x}L^{-1}(\xi) + w_3\mathbf{c}_l^R\mathbf{x} + w_4 \gamma_l^R\mathbf{x}\mathbf{R}^{-1}(\xi)$, $l = 1, \ldots, k$; w_1, w_2, w_3, w_4 are nonnegative weights, e. g. $w_1 = w_3 = 0.3$, $w_2 = w_4 = 0.2$. The deterministicobjectives (19) are linear even if reference functions L and R are nonlinear.

There exist approaches proposing a substitution of each objective by several deterministic objectives corresponding to extreme values of several ξ-level sets [9,28].

Finally, let us mention a comparison technique of fuzzy numbers, which is based on the compensation of area determined by the membership functions of two fuzzy numbers being compared. This technique, which has been characterized in [17] and [5], and then in [31] and [15], can be used directly to transform the comparison of fuzzy left- and right-hand side of the constraints, and of the fuzzy objectives and fuzzy goals into nonparametric deterministic equivalents. Although this technique seems intuitive, it has a convincing theoretical foundation.

Indeed, the semantics of fuzzy numbers considered in the MOLP problem with fuzzy coefficients is related to the representation of incomplete or vague states of information under the form of possibility distributions. This view of fuzzy numbers is concordant with the Dempster interpretation of fuzzy numbers as imprecise probability distributions [10]. In this perspective, the comparison of two fuzzy numbers can be substituted by the comparison of their mean values defined consistently with the well-known definition of expectation in probability theory. The idea exploited in [14] relies on the mathematical fact that, with respect to a fuzzy number, the possibility measure corresponds to an upper probability distribution, while the necessity measure, to a lower probability distribution of the corresponding random variable. Then it is reasonable to define the mean value of a fuzzy number as a closed interval whose bounds are expectations of upper and lower probability distributions. The comparison of two fuzzy numbers boils down to the comparison if arithmetic means of these bounds, which is computationally equivalent to the above mentioned technique based on area compensation, as shown in [15].

In consequence of application of all these comparison techniques, the MOLP problem with fuzzy coefficients is transformed to an associate deterministic MOLP problem, as (19), (18), (3) above, which should, preferably, be solved by one of existing interactive procedures (see, e. g., [43]).

Flexible MOLP with Fuzzy Coefficients

This problem combines the two semantics of fuzzy sets considered separately in flexible programming and in MOLP with fuzzy coefficients. This means that in addition to fuzzy coefficients in the objective functions and on the both sides of the constraints, the degree of satisfaction of fuzzy constraints and fuzzy goals is considered in fuzzy set terms.

A crucial question which has to be answered while solving a flexible MOLP problem with fuzzy coefficients is how to express the minimal conditions on the satisfaction of fuzzy constraints in deterministic terms.

In most of existing approaches, the minimal conditions on the satisfaction of fuzzy constraints (2) are expressed by one or two deterministic linear constraints which substitute the original fuzzy constraints. To give an idea of these crisp surrogates, let us present them in common terms from the most pessimistic to the most optimistic attitude. We assume the following form of the fuzzy left- and right-hand side of the ith constraint:

$$\begin{aligned}&\widetilde{\mathbf{a}}_i\mathbf{x} = (\mathbf{a}_i^L\mathbf{x}, \mathbf{a}_i^R\mathbf{x}, \alpha_i^L\mathbf{x}, \alpha_i^R\mathbf{x})_{LR},\\ &\widetilde{b}_i = (b_i, 0, \beta_i)_{LR},\end{aligned}$$

a) (see [3,40])

$$\mathbf{a}_i^R\mathbf{x} + \alpha_i^R\mathbf{x}R^{-1}(\rho) \leq b_i, \quad \rho \in [0,1];$$

b) (see [22,25,44])

$$\begin{cases}\mathbf{a}_i^R\mathbf{x} \leq b_i\\ \mathbf{a}_i^R\mathbf{x} + \alpha_i^R\mathbf{x}R^{-1}(\varepsilon) \leq b_i + \beta_i R^{-1}(\varepsilon),\\ \quad \varepsilon \in [0,1];\end{cases}$$

c) (see [4])

$$\begin{aligned}&\mathbf{a}_i^R\mathbf{x} + \alpha_i^R\mathbf{x}R^{-1}(\sigma) \leq b_i + \beta_i R^{-1}(\sigma),\\ &\sigma \in [0,1];\end{aligned}$$

d) (see [8,34,35])

$$\begin{cases}\mathbf{a}_i^L\mathbf{x} - b_i \leq \alpha_i^L\mathbf{x}L^{-1}(\tau) + \beta_i R^{-1}(\tau),\\ \tau \in [0,1], \text{ optimistic},\\ \mathbf{a}_i^R\mathbf{x} + \alpha_i^R\mathbf{x}R^{-1}(\eta) \leq b_i + \beta_i R^{-1}(\eta),\\ \eta \in [0,1], \text{ pessimistic};\end{cases}$$

e) (see [23])

$$\begin{cases} \mathbf{a}_i^R\mathbf{x} \le b_i + \delta\beta_i, \\ \delta + \varepsilon \in [0,1],\ \delta \ge 0,\ \varepsilon \ge 0, \\ \mathbf{a}_i^R\mathbf{x} + (1-\varepsilon-\delta)\alpha_i^R\mathbf{x} \le b_i + (1-\varepsilon)\beta_i; \end{cases}$$

f) (see [33,19])

$$\mathbf{a}_i^L\mathbf{x} - \alpha_i^L\mathbf{x}L^{-1}(\alpha) \le b_i + \beta_i R^{-1}(\alpha),$$
$$\alpha \in [0,1].$$

In all these approaches, the parameters α, δ, ε, η, τ, ρ, σ can be used by the decision maker to control the degree of satisfaction of fuzzy constraints in an interactive way.

Figure 1 shows results of conditions a)–f) applied on a common fuzzy constraint. Although it is the case in Fig. 1, the reference functions L and R need not be linear in the above conditions.

Another interpretation of fuzzy constraints has been given in [24]. The ith fuzzy constraint is replaced by the pessimistic condition proposed in [34] and by a new objective:

$$\mathbf{a}_i^R\mathbf{x} + \alpha_i^{R\varepsilon}\mathbf{x} \le b_i + \beta_i^{\varepsilon}, \tag{20}$$

$$\mu_i(\mathbf{x}) \to \max, \tag{21}$$

where membership function $\mu_i(\mathbf{x})$ is defined according to (4). More detailed discussion of the interpretation of fuzzy constraints can be found in [30].

If fuzzy goals are specified as L-R fuzzy numbers $\widetilde{g}_l = (g_l, 0, \nu_l)_{LL}$ ($l = 1, \dots, k$), then the satisfying conditions

$$\widetilde{\mathbf{c}}_l\mathbf{x} \widetilde{\le} \widetilde{g}_l, \quad l = 1, \dots, k, \tag{22}$$

can be treated as additional fuzzy constraints. In accordance to the chosen interpretation of the fuzzy inequality relation, (22) can be substituted by one or two crisp inequalities listed above or by (20) and (21). Another proposal has been made by R. Slowinski in [34,35]; the degree of satisfaction of fuzzy goals is represented there by the levels of intersection of left reference functions of $\widetilde{\mathbf{c}}_l\mathbf{x}$ with right reference functions of g_l ($l = 1, \dots, k$):

$$L\left(\frac{\mathbf{c}_l^L\mathbf{x} - g_l}{\gamma_l^L\mathbf{x} + \nu_l}\right) \to \max, \quad l = 1, \dots, k. \tag{23}$$

These crisp objectives substitute the fuzzy ones. In the case of linear reference functions L, functions (23) become linear fractional:

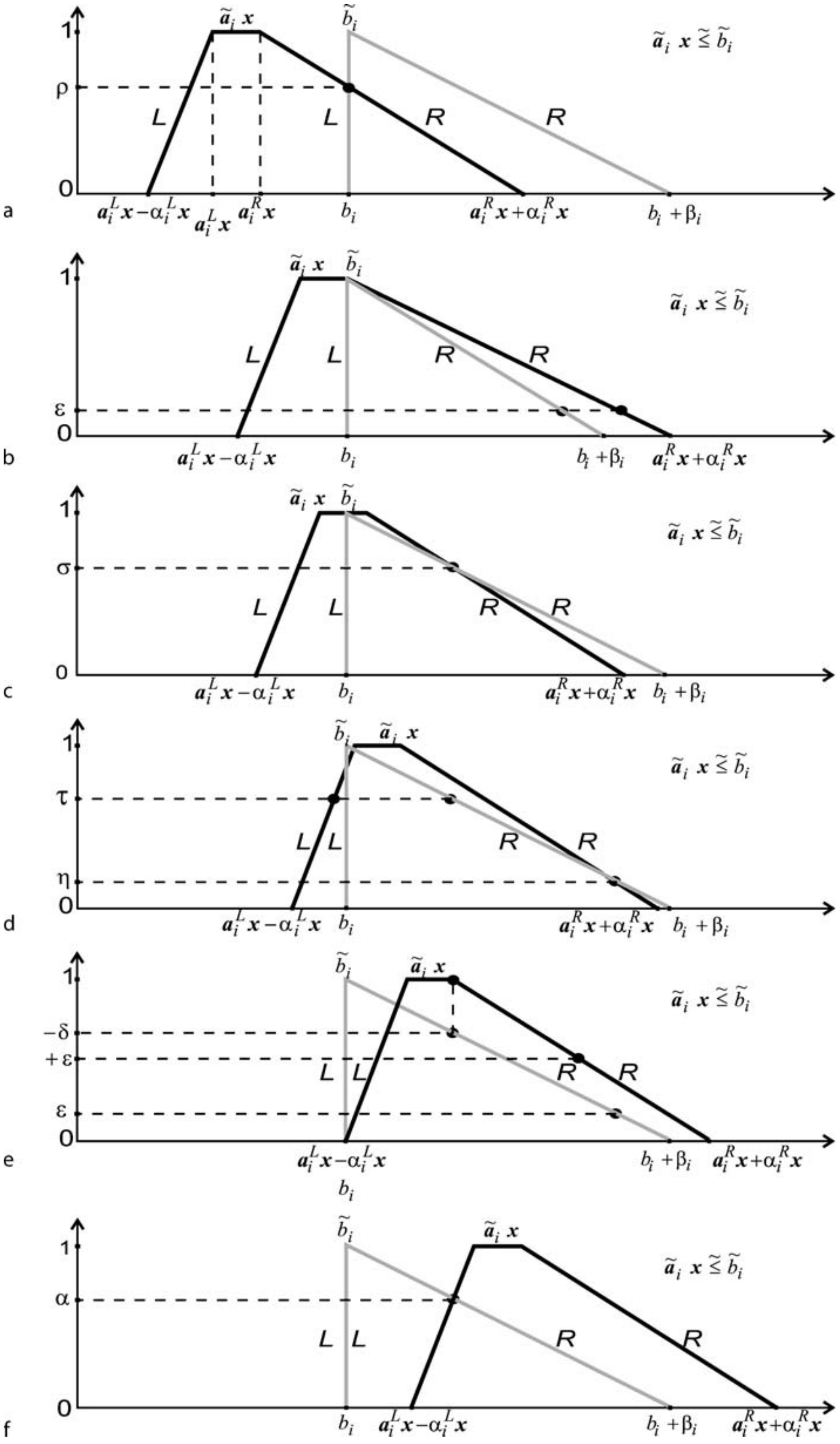
$$\frac{\mathbf{c}_l^L\mathbf{x} - g_l}{\gamma_l^L\mathbf{x} + \nu_l} \to \min, \quad l = 1, \dots, k. \tag{24}$$

The crisp objectives (24) and the optimistic and pessimistic conditions d) on the satisfaction of fuzzy constraints have been used in the FLIP method presented in [8,34,35,39]. They constitute an associate deterministic multi-objective *linear-fractional programming* (MOLFP) problem. In FLIP, the MOLFP problem is solved using an *interactive sampling procedure*. In each calculation step of this procedure, a sample of nondominated points (*Pareto optimal solutions*) of the MOLFP problem is generated and then shown to the decision maker who is asked to select the one that fits best his/her preferences. If the selected point is not the final compromise, it becomes a central point of a nondominated region that is sampled in the next calculation step. In this way, the sampled part of the nondominated set is successively reduced (focusing phenomenon) until the most satisfactory efficient point (compromise solution) is reached. An important advantage of the method presented above is that the only optimization procedure to be used is a linear programming one. Moreover, it has a simple scheme and allows retractions to the points abandoned in previous iterations.

The interaction with the decision maker takes place at two levels: first when fixing the safety parameters and then in the course of the guided generation and evaluation of the nondominated points of the MOLFP problem.

Let us precise that the fuzzy goals g_l ($l = 1, \dots, k$) do not influence the set of nondominated points of the MOLFP problem; they rather play the role of a visual reference than that of a preferential information influencing the set of generated proposals for the compromise solution.

An important feature of any software implementing a fuzzy multi-objective programming method is the presentation of candidate solutions in the interactive process. In the FLIP software, the Pareto optimal solutions of the MOLFP problem are shown not only numerically but also graphically, in terms of mutual positions of fuzzy numbers corresponding to original objectives and aspiration levels on the one hand, and to



Fuzzy Multi-objective Linear Programming, Figure 1
Results of conditions a)–f) applied on a common fuzzy constraint

left- and right-hand side of original constraints on the other hand [6]. In this way, the decision maker gets quite a complete idea of the quality of each proposed solution.

The quality is evaluated taking into account the following characteristics:

- scores of fuzzy objectives in relation to the goals;
- dispersion of values of the fuzzy objectives due to uncertainty;
- safety of the solution or, using a complementary term, the risk of violation of the constraints.

So, the definition of the best compromise involves not only the scores on particular objectives but also the safety of the corresponding solution. It is possible due to *visual interaction* that needs graphical display of objectives and constraints for any analyzed solution. The comparison of fuzzy left- and right-hand side of the constraints, as well as evaluation of dispersion of the values of objectives, is practically infeasible on the basis of numerals only. The graphical presentation of proposed solutions is not only a 'user friendly' interface but the best way for a complete characterization of these solutions.

There exists an implementation of FLIP in Visual Basic in the MS-Excel environment; it allows a user to define all safety parameters and the parameter p of the Yager's formula (16) for the aggregation of fuzzy objectives and of fuzzy left-hand sides of fuzzy constraints. The candidates for the best compromise solution are displayed there both numerically and graphically.

Conclusions

Fuzzy multi-objective linear programming methods have often been proposed in view of specific applications (see, e. g., [6,18,30,34,39,44]). This means that the many proposals described in this article are based on different assumptions that are verified in different practical situations. The choice of a procedure for an actual decision problem should take into account these assumptions. In any case, the interactive process should enable the best use of the decision maker's knowledge of the problem. Fuzzy multi-objective linear programming can also be seen as a tool for an interactive *robustness analysis* of MOLP problems. It gives an insight into sensitivity of proposed solutions on changes of particular coefficients within some intervals and on changes of preferences as to degrees of satisfaction of the constraints.

See also

- ▶ Bi-objective Assignment Problem
- ▶ Decision Support Systems with Multiple Criteria
- ▶ Estimating Data for Multicriteria Decision Making Problems: Optimization Techniques
- ▶ Financial Applications of Multicriteria Analysis
- ▶ Multicriteria Sorting Methods
- ▶ Multi-objective Combinatorial Optimization
- ▶ Multi-objective Integer Linear Programming
- ▶ Multi-objective Optimization and Decision Support Systems
- ▶ Multi-objective Optimization: Interaction of Design and Control
- ▶ Multi-objective Optimization: Interactive Methods for Preference Value Functions
- ▶ Multi-objective Optimization: Lagrange Duality
- ▶ Multi-objective Optimization: Pareto Optimal Solutions, Properties
- ▶ Multiple Objective Programming Support
- ▶ Outranking Methods
- ▶ Portfolio Selection and Multicriteria Analysis
- ▶ Preference Disaggregation
- ▶ Preference Disaggregation Approach: Basic Features, Examples From Financial Decision Making
- ▶ Preference Modeling

References

1. Behringer FA (1981) A simplex-based algorithm for the lexicographically extended linear maxmin problem. Europ J Oper Res 7:274–283
2. Bellman RE, Zadeh LA (1970) Decision making in a fuzzy environment. Managem Sci 17(4):141–164
3. Buckley JJ (1988) Possibilistic linear programming with triangular fuzzy numbers. Fuzzy Sets and Systems 26:135–138
4. Carlsson C, Korhonen P (1986) A parametric approach to fuzzy linear programming. Fuzzy Sets and Systems 20:17–30
5. Chanas S (1987) Fuzzy optimization in networks. In: Kacprzyk J, Orlowski SA (eds) Optimization Models Using Fuzzy Sets and Possibility Theory. Reidel, London, pp 303–327
6. Czyzak P (1990) Application of "FLIP" method to farm structure optimization under uncertainty. In: Slowinski R,

Teghem J (eds) Stochastic versus Fuzzy Approaches to Multiobjective Mathematical Programming under Uncertainty. Kluwer, Dordrecht, pp 263–278

7. Czyzak P, Slowinski R (1991) "FLIP"-multiobjective fuzzy linear programming software with graphical facilities. In: Fedrizzi M, Kacprzyk J, Roubens M (eds) Interactive Fuzzy Optimization. Springer, Berlin, pp 168–187
8. Czyzak P, Slowinski R (1993) A visual interactive method for MOLP problems with fuzzy coefficients. In: Lowen R, Roubens M (eds) Fuzzy Logic-State of the Art. Kluwer, Dordrecht, pp 321–332
9. Delgado M, Verdegay JL, Vila MA (1989) A general model for fuzzy linear programming. Fuzzy Sets and Systems 29:21–30
10. Dempster AP (1967) Upper and lower probabilities induced by a multivalued mapping. Ann Math Statist 38:325–339
11. Dubois D, Fortemps Ph (1999) Computing improved optimal solutions to max-min flexible constraint satisfaction problems. Europ J Oper Res 118:95–126
12. Dubois D, Prade H (1980) Systems of linear fuzzy constraints. Fuzzy Sets and Systems 3:37–48
13. Dubois D, Prade H (1987) Fuzzy numbers-an overview. In: Bezdek JC (ed) Analysis of Fuzzy Information: vol 1, Mathematics and Logic. CRC Press, Boca Raton, FL, 3–39
14. Dubois D, Prade H (1987) The mean value of a fuzzy number. Fuzzy Sets and Systems 24:279–300
15. Fortemps Ph, Roubens M (1996) Ranking and defuzzification methods based on area compensation. Fuzzy Sets and Systems 82:319–330
16. Inuiguchi M, Ichihashi H, Tanaka H (1990) Fuzzy programming: A survey of recent developments. In: Slowinski R, Teghem J (eds) Stochastic versus Fuzzy Approaches to Multiobjective Mathematical Programming under Uncertainty. Kluwer, Dordrecht, pp 45–68
17. Kolodziejczyk W (1986) Orlovsky's concept of decision-making with fuzzy preference relation-further results. Fuzzy Sets and Systems 19:11–20
18. Lai Y-J, Hwang C-L (1994) Fuzzy multiple objective decision making: Methods and applications. Springer, Berlin
19. Luhandjula MK (1987) Multiple objective programming with possibilistic constraints. Fuzzy Sets and Systems 21:135–146
20. Luhandjula MK (1989) Fuzzy optimization: An appraisal. Fuzzy Sets and Systems 30:257–282
21. Negoita CV (1981) The current interest in fuzzy optimization. Fuzzy Sets and Systems 6:261–269
22. Ramik J, Rimanek J (1985) Inequality between fuzzy numbers and its use in fuzzy optimization. Fuzzy Sets and Systems 16:123–138
23. Ramik J, Rommelfanger H (1993) A single-and a multi-valued order on fuzzy numbers and its use in linear programming with fuzzy coefficients. Fuzzy Sets and Systems 57:203–208
24. Rommelfanger H (1988, 1994) Entscheiden bei Unschaerfe-Fuzzy Decision Support Systeme. Springer, Berlin
25. Rommelfanger H (1989) Inequality relations in fuzzy constraints and their use in linear fuzzyoptimization. In: Verdegay JL, Delgado M (eds) The Interface between Artificial Intelligence and Operational Research in Fuzzy Environment. TÜV Rheinland, Köln, 195–211
26. Rommelfanger H (1990) "FULPAL"-an interactive method for solving (multi-objective) fuzzy linear programming problems. In: Slowinski R, Teghem J (eds) Stochastic versus Fuzzy Approaches to Multiobjective Mathematical Programming under Uncertainty. Kluwer, Dordrecht, pp 279–299
27. Rommelfanger H (1996) Fuzzy linear programming and applications. Europ J Oper Res 92:512–527
28. Rommelfanger H, Hanuscheck R, Wolf J (1989) Linear programming with fuzzy objectives. Fuzzy Sets and Systems 29:31–48
29. Rommelfanger H, Keresztfalvi T (1991) Multi-criteria fuzzy optimization based on Yager's parametrized t-norm. Found Computing and Decision Sci 16:99–110
30. Rommelfanger H, Slowinski R (1998) Fuzzy linear programming with single or multiple objective functions. In: Slowinski R (ed) Fuzzy Sets in Decision Analysis: Operations Research and Statistics. Kluwer, Dordrecht, pp 179–213
31. Roubens M (1990) Inequality constraints between fuzzy numbers and their use in mathematical programming. In: Slowinski R, Teghem J (eds) Stochastic versus Fuzzy Approaches to Multiobjective Mathematical Programming under Uncertainty. Kluwer, Dordrecht, pp 321–330
32. Sakawa M (1993) Fuzzy sets and interactive multiobjective optimization. Plenum, New York
33. Sakawa M, Yano H (1989) Interactive fuzzy satisficing method for multiobjective nonlinear programming problems with fuzzy parameters. Fuzzy Sets and Systems 30:221–238
34. Slowinski R (1986) A multicriteria fuzzy linear programming method for water supply system development planning. Fuzzy Sets and Systems 19:217–237
35. Slowinski R (1990) "FLIP"-an interactive method for multiobjective linear programming with fuzzy coefficients. In: Slowinski R, Teghem J (eds) Stochastic versus Fuzzy Approaches to Multiobjective Mathematical Programming under Uncertainty. Kluwer, Dordrecht, pp 249–262
36. Slowinski R (1997) Fuzzy multi-objective linear programming. Belgian J Oper Res Statist Comput Sci 37:83–98
37. Slowinski R (1997) Interactive fuzzy multiobjective programming. In: Climaco J (ed) Multicriteria analysis. Springer, Berlin, 202–212
38. Slowinski R, Teghem J (1988) Fuzzy versus stochastic approaches to multicriteria linear programming under uncertainty. Naval Res Logist 35:673–695. Reprinted in: Dubois D, Prade H, Yager R (eds) (1993) Readings in Fuzzy Sets

for Intelligent Systems. Morgan Kaufmann, San Mateo, CA, 810–821

39. Slowinski R, Teghem J (1990) Stochastic versus fuzzy approaches to multiobjective mathematical programming under uncertainty. Kluwer, Dordrecht
40. Tanaka H, Asai K (1984) Fuzzy linear programming with fuzzy numbers. Fuzzy Sets and Systems 13:1–10
41. Tanaka H, Okuda T, Asai K (1974) On fuzzy-mathematical programming. J Cybernetics 3(4):37–46
42. Vanderpooten D, Vincke Ph (1989) Description and analysis of some representative interactive multicriteria procedures. Math Comput Modelling, 12:1221–1238
43. Verdegay JL (1984) Application of fuzzy optimization in operational research. Control and Cybernetics 13:229–239
44. Wolf J (1988) Lineare Fuzzy Modelle zur Unterstuetzung der Investitionsentscheidung. Lang-Verlag, Frankfurt am Main
45. Zadeh LA (1965) Fuzzy sets. Inform and Control 8:338–353
46. Zimmermann HJ (1976) Description and optimization of fuzzy systems. Internat J General Syst 2:209–216
47. Zimmermann HJ (1978) Fuzzy programming and linear programming with several objective functions. Fuzzy Sets and Systems 1:45–55
48. Zimmermann HJ (1985) Applications of fuzzy sets theory to mathematical programming. Inform Sci 36:29–58