An Introduction to Submodular Functions and Optimization

Maurice Queyranne
University of British Columbia,
and IMA Visitor (Fall 2002)

IMA, November 4, 2002

1. Submodular set functions
- Generalization to lattice functions
- Why are submodular functions interesting?
- Example: joint replenishment functions in supply chain management
- Example: forests and trees
- Matroids rank functions (optional)
- Example: cut functions

2. Submodular polyhedra and greedy algorithms
- Submodular polyhedra
- Submodular polyhedron greedy algorithm
- Example: the core of convex games
- Example: scheduling polyhedra 3. Minimizing submodular functions
- Ellipsoid approaches
- The Edmonds-Cunningham approach
- Column generation
- A least squares approach
- Recent, more combinatorial algorithms
- Open questions

1. Submodular set functions:

Let N be a finite set (the *ground set*
2^N denotes the set of all subsets of N

A set function $f : 2^N \mapsto \mathbb{R}$ is
• *submodular* iff for all $A, B \subseteq N$

$$f(A \cup B) + f(A \cap B) \leq f(A) + f(B)$$

• *supermodular* iff $-f$ is submodular
• *modular* iff both sub- and supermodular

Generalization to lattice functions:

A *lattice* L is a a partially ordered set in which any two elements $a, b \in L$ have
• a least common upper bound (*join*) $a \vee b$
and
• a largest common lower bound (*meet*) $a \wedge b$

A function $f : L \mapsto \mathbb{R}$ is *submodular* iff for all $a, b \in L$,

$$f(a \vee b) + f(a \wedge b) \leq f(a) + f(b)$$

Why are submodular functions interesting?

Role somewhat similar to that played by convex/concave functions in continuous optimization:

- arise in many applications
- preserved by some useful operations
- lead to nice theory and structural results
- some related optimization problems can be solved efficiently
- nice parametric and postoptimality properties (not discussed in this talk)

Lemma: When N is finite, the submodularity of $f : 2^N \mapsto \mathbb{R}$ is equivalent to:

• $\forall A \subset B \subset N,\ \forall j \in N \setminus B,$
$$f(A+j) - f(A) \geq f(B+j) - f(B)$$
(*nonincreasing first differences, economies of scale, economies of scope*),

and also to:

• $\forall A \subset N,\ \forall i, j \in N \setminus A,$
$$f(A+j) - f(A) \geq f(A+i+j) - f(A+i)$$
(*nonpositive second discrete differences, local submodularity*)

where $f(A+j)$ stands for $f(A \cup \{j\})$ (assuming $j \notin A$).

Corollary (cardinality functions):
If $f(A) = g(|A|)$ for all $A \subseteq N$ where $g : \mathbb{N} \mapsto \mathbb{R}$ then f is submodular iff g is concave

Example: joint replenishment functions in supply chain management

Ground set N is the set of all items being stocked (*stock keeping units*, or *sku*s)

Managing the inventory of these items implies finding an optimum tradeoff between
- ordering costs and economies of scale (which favor large, infrequent replenishments);

and
- holding costs and service requirements (which favor small, frequent replenishments)

Ordering costs often include *fixed costs*, which depend only on the subset of jointly ordered items (and not their order quantities).

Economies of scope in
- procurement,
- order processing,
- transportation,

often imply that these joint replenishment fixed costs are submodular set functions.

Example: Forests and Trees

In a given graph $G = (V, E)$, a subgraph $S = (V, F)$ is a *forest* iff it contains no cycle.

A *(spanning) tree* is a forest (V, F) with $|F| = |V| - 1$

Graphic rank function:
- ground set $N = E$
- the *rank* of $A \subseteq E$ is

$r(A) = \max\{|F| : F \subseteq A$ and (V, F) is a forest$\}$

Lemma: This rank function is submodular.

This graphic rank function arises in formulating *connectivity constraints* for *network design problems*

The *number of connected components* in the subgraph (V, A) is: $nc(A) = |V| - r(A)$
$\Rightarrow$ it is a supermodular function

Matroid rank functions (optional material)

A *set system* $(N, \mathcal{F})$ is defined by
- a finite ground set N, and
- a family $\mathcal{F} \subseteq 2^N$ of *independent sets*.

A set system $(N, \mathcal{F})$ is a *matroid* iff
(M1) $\emptyset \in \mathcal{F}$, and
(M2) if $X \subseteq Y \in \mathcal{F}$ then $X \in \mathcal{F}$
(M3) if $X, Y \in \mathcal{F}$ and $|X| > |Y|$
then $\exists e \in X \setminus Y$ such that $Y + e \in \mathcal{F}$

Define the *rank* function of a set system $(N, \mathcal{F})$:

$$\forall X \subseteq N,\ r(X) = \max\{|F| : X \supseteq F \in \mathcal{F}\}$$

Theorem: Let $r : 2^N \mapsto \mathbb{N}$.
The following are equivalent:
(i) $\mathcal{F} = \{F \subseteq N : r(F) = |F|\}$ defines a matroid $(N, \mathcal{F})$ and r is its rank function
(ii) r satisfies, for all $X, Y \subseteq N$
(R1) $r(X) \leq |X|$;
(R2) if $X \subseteq Y$ then $r(X) \leq r(Y)$; and
(R3) r is submodular.

Example: Cut functions
Let $G = (V, A)$ be a digraph,
with given arc *capacities* $c(a)$, $\forall a \in A$
Here, the ground set $N = V$.

- the *cut* (or *coboundary*) $\delta^+(S)$ of $S \subseteq V$ is:
  $\delta^+(S) = \{a = (i, j) \in A : i \in S \text{ and } j \notin S\}$

- *global cut function* $f : 2^V \mapsto \mathbb{R}$ is defined by
  $f(S) = c(\delta^+(S)) = \sum_{a \in \delta^+(S)} c(a) \quad \forall S \subseteq V$

- given $s \neq t \in V$, the s,t*-cut function*
  $f_{st} : 2^{V_{st}} \mapsto \mathbb{R}$ is defined by
  $f_{st}(S) = f(S + s) \quad \forall S \subseteq V_{st} = V \setminus \{s, t\}$

Lemma: If $c \geq 0$ then these cut functions f and f_{st} are submodular.

Proof: $f(S) + f(T) - f(S \cap T) - f(S \cup T)$
$= c(A(S : T)) + c(A(T : S))$
where $A(X : Y) = \{(i, j) \in A : i \in X,\ j \in Y\}$.
QED

All these also apply to the undirected case.

Cut functions arise in connection with many combinatorial optimization models, in particular:

- network flows
- optimum selection of contingent investments
- design of an open pit mine
- precedence-constrained scheduling

Cut functions are also used in formulating connectivity constraints, in particular for

- vehicle routing (e.g., TSP)
- network design

2. Submodular polyhedra and Greedy algorithms

Given a finite set N and $f : 2^N \mapsto \mathbb{R}$, let

$$P(f) = \{x \in \mathbb{R}^N : x(A) \leq f(A)\ \forall A \subseteq N\}$$

where $x(A) = \sum\{x(e) : e \in A\}$.

Remark: $P(f) \neq \emptyset$ iff $f(\emptyset) \geq 0$

Define $P(f)$ to be a *submodular polyhedron* iff f is submodular.

A linear programming problem:
Given:
- a finite set N,
- a set function $f : 2^N \mapsto \mathbb{R}$ with $f(\emptyset) \geq 0$, and
- a weight vector $w \in \mathbb{R}^N$,

solve

$$\text{(LP)} \quad \max\{wx : x \in P(f)\}$$

Remark: (LP) has bounded optimum iff $w \geq 0$.

Submodular Polyhedron Greedy Algorithm

1. Sort the elements of N as
$$w(e_1) \geq w(e_2) \geq \cdots \geq w(e_n)$$
2. Let $V_0 = \emptyset$.
   For $i = 1$ to n let
   $V_i = V_{i-1} + e_i$ and $x^G(e_i) = f(V_i) - f(V_{i-1})$.

Theorem (Edmonds, 1971; Lovász, 1983): The Submodular Polyhedron Greedy Algorithm solves (LP) for all weight vectors $w \geq 0$ iff f is submodular.

- If, in addition, f is integral, then the greedy solution x^G is integral

- An optimal dual solution is
$$y(A) = \begin{cases} w(e_i) - w(e_{i+1}) & \text{if } A = V_i, \\ 0 & \text{otherwise} \end{cases}$$
  where $w(e_{n+1}) = 0$.

$\Rightarrow$ when f is submodular, the constraints defining $P(f)$ are *totally dual integral*.

Corollary:
When f is submodular, the *base polytope*

$$B(f) = P(f) \cap \{x \in \mathbb{R}^N : x(N) = f(N)\}$$

is non-empty.

Example: The core of convex games
Cooperative games:
cost (or profit) allocation example:
- how to share the cost of (or profit from) a facility serving different "players"?

A (*characteristic-function*) *game* is a pair (N, f) where:
- N is a set of *players*
- $f : 2^N \mapsto \mathbb{R}$ is a set function such that $f \geq 0$ and $f(\emptyset) = 0$

- a subset $S \subset N$ ($S \neq \emptyset$) is a possible *coalition*
- $f(S)$ is the cost if the facility is built (and/or operated) only for the players in S
- N is the *grand coalition*

- A game is *convex* if f is submodular

A *cost allocation* is a vector $x \in \mathbb{R}^N$ such that

$x(N) = f(N)$.

It is *efficient* (or *stable*) if no coalition has any advantage to seceding:

$x(S) \le f(S)$ for all $S \subset N$

(*subgroup rationality*; the special case where $S = \{i\}$ is *individual rationality* for player i)

The *core* is the set of all efficient allocations:

$$\begin{aligned} \text{core}(N, f) &= \{x \in \mathbb{R}^N : x(N) = f(N), \\ &\qquad x(S) \le f(S)\ \forall S \subset N\} \\ &= B(f) \end{aligned}$$

Theorem (Shapley, 1971)

- The core of a convex game is a nonempty polytope.
- Its extreme points are obtained by the Submodular Polyhedron Greedy Algorithm.

Many other related *solution concepts* exist.

Example: Scheduling Polyhedra

Given a machine, which can process at most one task at a time; and
n jobs $J_1, \ldots, J_n$ to be processed, each

- available at date 0,
- with given processing time $p_j > 0$

(and possibly other constraints)

A feasible schedule S induces a vector

$$C^S = (C_1^S, C_2^S, \ldots, C_n^S) \in \mathbb{R}^N$$

where C_j is the completion time of job J_j in S.

The *performance region* is the set of all such feasible completion time vectors C^S.

- a non-connected set (e.g., without additional constraints, it is the union of $n!$ disjoint affine cones)

We are often interested in minimizing a linear (or concave) function of C^S
$\Rightarrow$ it suffices to consider the *convex hull* of the performance region.

The ground set is the job set N.
For $A \subseteq N$, define

$$g(A) = \frac{1}{2}\left(\sum_{j \in A} p_j\right)^2 + \frac{1}{2}\sum_{j \in A} p_j^2$$

Lemma: g is a supermodular set function

Theorem (Wolsey, 1985; Q, 1993):
(i) The supermodular polyhedron

$$Q(g) = \{C \in \mathbb{R}^N : \sum_{j \in A} p_j C_j \geq g(A) \text{ for all } A \subseteq N\}$$

is the convex hull of the performance region.

(ii) Its base polytope

$$Q(g) \cap \{C \in \mathbb{R}^N : \sum_{j \in N} p_j C_j = g(N)\}$$

is that of the performance region restricted to all feasible schedules without idle time.

Remark: These results apply to both *preemptive* and *nonpreemptive* schedules.

3. Minimizing Submodular Functions

$$\text{minimize}\{f(S) : S \subseteq N\}$$

where $f : 2^N \mapsto \mathbb{R}$ is submodular

- equivalently, we may want to *max*imize a *super*modular function

- w.l.o.g., we may assume that f is *normalized* i.e., $f(\emptyset) = 0$.

Example: the *separation problem* for a submodular polyhedron

Given
- a submodular set function $f : 2^N \mapsto \mathbb{R}$
- a point $\overline{x} \in \mathbb{R}^N$

decide whether or not $\overline{x} \in P(f)$
and, if not, find a violated inequality.

A *most violated* inequality is defined by $S \subseteq N$ maximizing the *violation* $v_{\overline{x}}(x) = \overline{x}(S) - f(S)$
- $v_{\overline{x}}$ is a supermodular function

How to input a submodular set function?

Assume that f is given by a *value oracle*, i.e., a "black box" which, for any input subset A returns the value $f(A)$

• let FE be an upper bound on the time needed for such a function evaluation

Theorem (Grötschel, Lovász & Schrijver, 1981)
A submodular set function can be minimized in *strongly polynomial time*
(i.e., in time polynomial in $|N|$ and FE)

Remark: related problems can be difficult, e.g.,
- *max*imizing a submodular function f, or
- minimizing a submodular function f, subject to a cardinality constraint, such as

  $|A| = b$, or $|A| \geq b$, or $|A| \leq b$

are NP-hard when f is a cut function.
Applications: • facilities location
• entropy maximization

Extensions

A submodular function on a *product lattice* (or a sublattice thereof) can be minimized using a *pseudopolynomial* number of submodular set function minimizations (Q and Tardella, 1992)

- pseudopolynomial is best possible for such problems

The *discrete submodular resource allocation problem* with separable concave profit function,

$$\text{(RA)} \qquad \max\{\sum_{j=1}^{n} g_j(x_j) : x \in P(f) \cap \mathbb{N}^N\}$$

can be solved in polynomial time (Federgruen and Groenevelt, 1986)

Other types of *discretely convex functions* can also be minimized in polynomial time (Favati and Tardella; Murota, Shioura)

Ellipsoid approaches
(Grötschel, Lovász & Schrijver, 1981, 1988)

1. (Polynomial equivalence of Optimization and Separation problems):
The Optimization problem on submodular polyhedra can be solved in P-time (Greedy Alg.)
$\Rightarrow$ the Separation problem can be solved in P-time (by the Ellipsoid method)

Let $\alpha \leq 0$ be a *trial value* for $f(S^*) = \min f(S)$ and define $f_\alpha : 2^N \mapsto \mathbb{R}$ by

$$f_\alpha(A) = \begin{cases} f(A) - \alpha & \text{if } A \neq \emptyset; \\ 0 & \text{otherwise} \end{cases}$$

Lemma: f_α is submodular when $\alpha \leq 0$

- Solve the Separation problem for $P(f_\alpha)$ with $\overline{x} = 0$. (Note: $0 \in P(f_\alpha)$ iff $f(S^*) \geq \alpha$)

- Use binary search for the value $\alpha = f(S^*)$ over an interval $[LB,\ 0]$ where LB is any lower bound on $f(S^*)$ e.g., $LB = \sum_e [f(N) - f(N \setminus \{e\})]$

2. Lovász's extension $\hat{f}$ of f to $\mathbb{R}^N_+$:

$\hat{f}(w) = \max\{wx : x \in P(f)\}$ for all $w \in \mathbb{R}^N,\ w \geq 0\}$

(Note: non-standard definition!)

For any set function f, the function $\hat{f}$

- is piecewise linear convex
- is (positively) homogeneous:

$$\hat{f}(\lambda w) = \lambda \hat{f}(w) \text{ for all } \lambda \geq 0\ (\lambda \in \mathbb{R})$$

Lemma: When f is submodular, $\hat{f}$

- is evaluated in P-time (Greedy Alg.)
- coincides with f at all 0-1 points:

$$f(S) = \hat{f}(\chi^S) \text{ for all } S \subseteq N$$

- has its minimum over the unit cube $[0,1]^E$ attained at a vertex

$\Rightarrow$ $\hat{f}$ can be minimized over the unit cube in P-time (by the Ellipsoid method)

3. Strongly P-time algorithm in GLS (1988) uses $O(|N|^4)$ f-function evaluations

How about a P-time *combinatorial* algorithm?

The Edmonds-Cunningham approach

Lemma: Let u^* be an optimum solution to the LP

$$
\begin{aligned}
z^* = \quad & \min \sum_{S \subseteq N} f(S)\, u_S \\
& \text{s.t.} \sum_{S : e \in S} u_S \leq 1 \quad \forall e \in N \\
& \qquad u \geq 0
\end{aligned}
$$

If f is submodular and normalized then
$S^* = \bigcup\{S : u^*_S > 0\}$ minimizes $f(S)$.

The dual problem is:

$$
\begin{aligned}
(S - LP) \qquad & \max \sum_{e \in N} x_e \\
& \text{s.t.} \quad x(S) \leq f(S) \quad \forall S \subseteq N \\
& \qquad\quad x \leq 0
\end{aligned}
$$

Cunningham (1984, 1985):

- a pseudopolynomial time algorithm polynomial in $|N|$ and $f(S^*)$

$\Rightarrow$ strongly P-time for matroid rank functions

Column generation approach

The extreme points $b^1, \ldots, b^k$ of $P(f)$ can be generated by the Greedy Algorithm, and

$$P(f) = \text{conv}\{b^1, \ldots, b^k\} + \mathbb{R}^N_-$$

Applying Dantzig-Wolfe decomposition:

$$
\begin{aligned}
(DW) \qquad\qquad & \max \sum_{u \in N} x_v \\
& \text{s.t. } \; x \leq \sum_{i=1}^{k} \lambda_i b^i \\
& \qquad\quad \sum_{i=1}^{k} \lambda_i = 1 \\
& \qquad\quad \lambda \geq 0, \; x \leq 0
\end{aligned}
$$

- the column generation subproblem is solved by the Greedy Algorithm
- "greedy" column generation is equivalent to Successive Linear Programming (SLP) applied to the Lovász extension problem
$$\min\{\widehat{f}(x) : 0 \leq x \leq 1\}$$

A least squares approach

Theorem (Fujishige, 1984)
If x^* solves the *least squares problem*

$$(LS) \qquad \max \sum_{e \in N} x_e^2$$
$$\text{s.t.} \quad x(S) \leq f(S) \quad \forall S \subseteq N$$
$$x \leq 0$$

then

- $f(S^*) = x^*(N)$

- S^* is the largest *tight set*
  i.e., the largest set S such that $x^*(S) = f(S)$

Recent, more combinatorial algorithms

Recent primal feasible algorithms for (S-LP):

- a convex representation of current solution x
- augmenting paths, and
- exchange capacities in strongly P-time

Schrijver (1999):

- "short" augm. paths in a layered network
- lower bounds on exchange capacities
- removing selected arcs from augm. network

Iwata, Fleischer and Fujishige (IFF, 1999-2000):

- scaling approach
- relaxation of f with a scaled penalty
- "fat" augmenting path analysis

(see Fleischer, OPTIMA 64, Sept. 2000; and McCormick's chapter to appear in Handbook of Discrete Optimization)

Iwata (2000):
a "fully combinatorial" variant of IFF
- uses only additions, subtractions and comparisons (no multiplications or divisions)
- O($(n^9 \log^2 n)FE + n^{11} \log^2 n$) running time

Iwata (2001): Hybrid algorithm
- combines ideas from Schrijver's algorithm into IFF
- O($n^8 FE + n^9$) running time

Open questions

How low can we reduce the running time?

Can we find (nontrivial) *lower bounds*
- on the running time of any submodular minimization algorithm?
- on the number of function evaluations needed to minimize a submodular function given by a value oracle?

A few selected references

The following textbooks have a chapter on submodular set functions and optimization:

B. Korte & J. Vygen, *Combinatorial Optimization: Theory and Algorithms*, Springer, 2002.

G.L. Nemhauser & L.A. Wolsey, *Integer and Combinatorial Optimization*, Wiley 1988, 1999.

Monographs on sub/supermodularity:

S. Fujishige, *Submodular Functions and Optimization*, North-Holland, 1991.

H. Narayanan, *Submodular Functions and Electrical Networks*, North-Holland, 1997.

D.M. Topkis, *Supermodularity and Complementarity*, Princeton Univ. Press, 2001.

Recent reviews on submodular set function minimization:

L. Fleischer, "Recent Progress in Submodular Function Minimization", *OPTIMA* **64** (September 2000) 1–11.
`http://www.ise.ufl.edu/~optima/optima64.pdf`

S.T. McCormick, "Submodular Function Minimization", to appear in *Handbook of Combinatorial Optimization*.
(`stmv@adk.commerce.ubc.ca`)

A recent application in statistical mechanics:

J.-Ch. Anglès d'Auriac, F. Iglói, M. Preissmann & A. Sebő, "Optimal cooperation and submodularity for computing Potts' partition functions with a large number of states", *J. Phys. A: Math. Gen.* **35** (2002) 6973–6983.