Constraint Optimization Problems

Todd Ebert

Outline

Finding Optimial Model Solutions

Constraint Optimization Problem

Finding Optimal Model Solutions

Constraint Optimization Problem

- A constraint optimization problem is a quadruple $P = (V, D, C, f)$, where $f : \mathcal{A}(V) \to \mathcal{R}$ is called the **objective function**.

Finding Optimal Model Solutions

Constraint Optimization Problem

- A constraint optimization problem is a quadruple $P = (V, D, C, f)$, where $f : \mathcal{A}(V) \rightarrow \mathcal{R}$ is called the **objective function**.
- An **optimal solution** s for P is an assignment over V that satisfies all constraints in C, and for which $f(s)$ is optimal (is either a maximum or minimum) over all such assignments. $f(s)$ is called the **optimal objective value**.

Finding Optimal Model Solutions

Constraint Optimization Problem

- A constraint optimization problem is a quadruple $P = (V, D, C, f)$, where $f : \mathcal{A}(V) \to \mathcal{R}$ is called the **objective function**.
- An **optimal solution** s for P is an assignment over V that satisfies all constraints in C, and for which $f(s)$ is optimal (is either a maximum or minimum) over all such assignments. $f(s)$ is called the **optimal objective value**.
- In this lecture we assume the objective is to maximize f.

The Iteration Method for Optimizing $P = (V, D, C, f)$

Let s_0 be an initial solution for $P = (V, D, C)$.
Let $L = f(s_0)$.
Let U be an upperbound for f.
Let $M = (L + U)/2$ be the midpoint for L and U.

//Add the constraint that $f(a)$ must be at least as great as M
Let $\hat{C} = C \cup \{f(a) \geq M\}$.

While there is still time to search
- Find a solution s for $(V, D, \hat{C})$.
- If no solution is found, then set $U = M$ and
  - $M = (L + M)/2$.
- Otherwise,
  - Set $L = f(s)$.
  - Set $M = (L + U)/2$.

Shortcomings of the Iteration Method

Compound Objective Functions

An objective function can have a complex structure, and be comprised of several sub-functions. We call such an objective function a **compound** function. An optimization algorithm should take advantage of this structure.

Shortcomings of the Iteration Method

Compound Objective Functions

An objective function can have a complex structure, and be comprised of several sub-functions. We call such an objective function a **compound** function. An optimization algorithm should take advantage of this structure.

Example: Finding a Maximum Clique for $G = (V, E)$

Shortcomings of the Iteration Method

Compound Objective Functions

An objective function can have a complex structure, and be comprised of several sub-functions. We call such an objective function a **compound** function. An optimization algorithm should take advantage of this structure.

Example: Finding a Maximum Clique for $G = (V, E)$

- **Variables.** Boolean variables x_v, for each $v \in V$.

Shortcomings of the Iteration Method

Compound Objective Functions

An objective function can have a complex structure, and be comprised of several sub-functions. We call such an objective function a **compound** function. An optimization algorithm should take advantage of this structure.

Example: Finding a Maximum Clique for $G = (V, E)$

- **Variables.** Boolean variables x_v, for each $v \in V$.
- **Constraints.** For all $u, v \in V$, if x_u and x_v are both set to true, then $(u, v) \in E$.

Shortcomings of the Iteration Method

Compound Objective Functions

An objective function can have a complex structure, and be comprised of several sub-functions. We call such an objective function a **compound** function. An optimization algorithm should take advantage of this structure.

Example: Finding a Maximum Clique for $G = (V, E)$

- **Variables.** Boolean variables x_v, for each $v \in V$.
- **Constraints.** For all $u, v \in V$, if x_u and x_v are both set to true, then $(u, v) \in E$.
- **Objective Function.** $f(a) = \sum_{v \in V} a(x_v)$, is the sum of the assignments to all the model variables.

Soft and Hard Constraints

Terminology

Soft and Hard Constraints

Terminology

- **Hard constraint:** any constraint that *must* be satisfied by a model solution.

Soft and Hard Constraints

Terminology

- **Hard constraint:** any constraint that *must* be satisfied by a model solution.
- **Soft constraint:** any constraint that need not be satisfied by a model solution, but whose satisfaction improves the overall **quality** of the soluton.

Soft and Hard Constraints

Terminology

- **Hard constraint:** any constraint that *must* be satisfied by a model solution.
- **Soft constraint:** any constraint that need not be satisfied by a model solution, but whose satisfaction improves the overall **quality** of the soluton.
- **Soft constraints as functions.** A soft constraint may be viewed as a function $f : \mathcal{A}(\text{var}(f)) \to \mathcal{R}$, where $f(a)$ indicates the degree to which f is contributing to the quality of a solution that assigns a to the variables of f.

Soft and Hard Constraints

Terminology

- **Hard constraint:** any constraint that *must* be satisfied by a model solution.
- **Soft constraint:** any constraint that need not be satisfied by a model solution, but whose satisfaction improves the overall **quality** of the soluton.
- **Soft constraints as functions.** A soft constraint may be viewed as a function $f : \mathcal{A}(\text{var}(f)) \to \mathcal{R}$, where $f(a)$ indicates the degree to which f is contributing to the quality of a solution that assigns a to the variables of f.
- **Soft Constraint Model/Problem.** $P = (V, D, C, F)$, where F is the set of soft constraints.

Examples of Soft Constraints

Academic Scheduling

Examples of Soft Constraints

Academic Scheduling

- **Two-day Teaching Schedule.** An instructor prefers to have a two-day teaching schedule.

Examples of Soft Constraints

Academic Scheduling

- **Two-day Teaching Schedule.** An instructor prefers to have a two-day teaching schedule.
- **Curriculum Availability.** A **curriculum** is a set of courses that a student of some level (e.g. first-semester Junior) should take together. Some courses are offered in multiple sections that allow for different ways for a student to attain the curriculum. The weight function counts the number of ways. For example, suppose the curriculum is course A and course B. Course A has two sections that have been assigned as MW morning and TTH evening. Course B has three sections, assigned as MW evening, TTH morning, and Friday morning. A soft constraint function might assign a vlaue of $2 \times 3 = 6$, since there are 6 ways that a student can attain the curriculum based on this assignment.

Examples of Soft Constraints

Maximum Clique for $G = (V, E)$

For each $v \in V$ the constraint $x_v =$ true is a soft constraint with $f(a) = 1$ if $a(x_v) = 1$, and $f(a) = 0$ otherwise.

Examples of Soft Constraints

Maximum Clique for $G = (V, E)$

For each $v \in V$ the constraint $x_v =$ true is a soft constraint with $f(a) = 1$ if $a(x_v) = 1$, and $f(a) = 0$ otherwise.

MaxSAT Problem

Given a constraint model $P = (V, D, C)$, find an assignment $a \in \mathcal{A}(V)$ that maximizes the number of satisfied constraints. As a soft-constraint model, $\hat{P} = (V, D, \emptyset, C)$, where $c(a) = 1$ iff a satisfies c, and 0 otherwise.

Examples of Soft Constraints: Combinatorial Bidding

Problem Definition

Examples of Soft Constraints: Combinatorial Bidding

Problem Definition

- **Auction Set.** $A = \{a_1, \ldots, a_n\}$ is a set of items to be auctioned.

Examples of Soft Constraints: Combinatorial Bidding

Problem Definition

- **Auction Set.** $A = \{a_1, \ldots, a_n\}$ is a set of items to be auctioned.
- **Bid Set.** $B = \{b_1, \ldots, b_m\}$ is the set of bids. For $i = 1, \ldots, m$, $b_i = (S_i, p_i)$, where $S_i \subseteq A$ is a subset of items, and p_i is the total bidding price for those items.

Examples of Soft Constraints: Combinatorial Bidding

Problem Definition

- **Auction Set.** $A = \{a_1, \ldots, a_n\}$ is a set of items to be auctioned.
- **Bid Set.** $B = \{b_1, \ldots, b_m\}$ is the set of bids. For $i = 1, \ldots, m$, $b_i = (S_i, p_i)$, where $S_i \subseteq A$ is a subset of items, and p_i is the total bidding price for those items.
- **Goal.** find a subset $\hat{B} \subseteq B$ of bids in such a way that no two bids in $\hat{B}$ overlap in items, and the sum of all bid prices is maximum with respect to all such subsets of nonoverlapping bids.

Examples of Soft Constraints: Combinatorial Bidding

Bidding Example

Examples of Soft Constraints: Combinatorial Bidding

Bidding Example

- **Auctioned items.** $A = \{1, \ldots, 8\}$

Examples of Soft Constraints: Combinatorial Bidding

Bidding Example

- **Auctioned items.** $A = \{1, \ldots, 8\}$
- **Bids.** $b_1 = (\{2, 3, 5, 7, 8\}, 7)$, $b_2 = (\{7, 8\}, 4)$, $b_3 = (\{1, 5, 8\}, 6)$, $b_4 = (\{1, 8\}, 5)$, $b_5 = (\{1, 2, 4, 5, 6\}, 8)$, $b_6 = (\{3, 4, 5\}, 7)$, $b_7 = (\{2, 4, 5, 7\}, 6)$, and $b_8 = (\{1, 4\}, 4)$.

Examples of Soft Constraints: Combinatorial Bidding

Bidding Example

- **Auctioned items.** $A = \{1, \ldots, 8\}$
- **Bids.** $b_1 = (\{2,3,5,7,8\},7)$, $b_2 = (\{7,8\},4)$, $b_3 = (\{1,5,8\},6)$, $b_4 = (\{1,8\},5)$, $b_5 = (\{1,2,4,5,6\},8)$, $b_6 = (\{3,4,5\},7)$, $b_7 = (\{2,4,5,7\},6)$, and $b_8 = (\{1,4\},4)$.
- **Optimal subset.** $\hat{B} = \{b_2, b_5\}$ for a total bid value of 4+8=12.

Examples of Soft Constraints: Combinatorial Bidding

Bidding Example

- **Auctioned items.** $A = \{1, \ldots, 8\}$
- **Bids.** $b_1 = (\{2,3,5,7,8\}, 7)$, $b_2 = (\{7,8\}, 4)$, $b_3 = (\{1,5,8\}, 6)$, $b_4 = (\{1,8\}, 5)$, $b_5 = (\{1,2,4,5,6\}, 8)$, $b_6 = (\{3,4,5\}, 7)$, $b_7 = (\{2,4,5,7\}, 6)$, and $b_8 = (\{1,4\}, 4)$.
- **Optimal subset.** $\hat{B} = \{b_2, b_5\}$ for a total bid value of 4+8=12.

Soft Constraint Model for the Bidding Problem

Examples of Soft Constraints: Combinatorial Bidding

Bidding Example

- **Auctioned items.** $A = \{1, \ldots, 8\}$
- **Bids.** $b_1 = (\{2, 3, 5, 7, 8\}, 7)$, $b_2 = (\{7, 8\}, 4)$, $b_3 = (\{1, 5, 8\}, 6)$, $b_4 = (\{1, 8\}, 5)$, $b_5 = (\{1, 2, 4, 5, 6\}, 8)$, $b_6 = (\{3, 4, 5\}, 7)$, $b_7 = (\{2, 4, 5, 7\}, 6)$, and $b_8 = (\{1, 4\}, 4)$.
- **Optimal subset.** $\hat{B} = \{b_2, b_5\}$ for a total bid value of 4+8=12.

Soft Constraint Model for the Bidding Problem

- Boolean variables $V = \{x_1, \ldots, m\}$, where x_i true means bid i is accepted.

Examples of Soft Constraints: Combinatorial Bidding

Bidding Example

- **Auctioned items.** $A = \{1, \ldots, 8\}$
- **Bids.** $b_1 = (\{2, 3, 5, 7, 8\}, 7)$, $b_2 = (\{7, 8\}, 4)$, $b_3 = (\{1, 5, 8\}, 6)$, $b_4 = (\{1, 8\}, 5)$, $b_5 = (\{1, 2, 4, 5, 6\}, 8)$, $b_6 = (\{3, 4, 5\}, 7)$, $b_7 = (\{2, 4, 5, 7\}, 6)$, and $b_8 = (\{1, 4\}, 4)$.
- **Optimal subset.** $\hat{B} = \{b_2, b_5\}$ for a total bid value of 4+8=12.

Soft Constraint Model for the Bidding Problem

- Boolean variables $V = \{x_1, \ldots, m\}$, where x_i true means bid i is accepted.
- C: for all $1 \leq i < j \leq n$, $x_i \wedge x_j \rightarrow S_i \cap S_j = \emptyset$.

Examples of Soft Constraints: Combinatorial Bidding

Bidding Example

- **Auctioned items.** $A = \{1, \ldots, 8\}$
- **Bids.** $b_1 = (\{2,3,5,7,8\}, 7)$, $b_2 = (\{7,8\}, 4)$, $b_3 = (\{1,5,8\}, 6)$, $b_4 = (\{1,8\}, 5)$, $b_5 = (\{1,2,4,5,6\}, 8)$, $b_6 = (\{3,4,5\}, 7)$, $b_7 = (\{2,4,5,7\}, 6)$, and $b_8 = (\{1,4\}, 4)$.
- **Optimal subset.** $\hat{B} = \{b_2, b_5\}$ for a total bid value of 4+8=12.

Soft Constraint Model for the Bidding Problem

- Boolean variables $V = \{x_1, \ldots, m\}$, where x_i true means bid i is accepted.
- C: for all $1 \leq i < j \leq n$, $x_i \wedge x_j \rightarrow S_i \cap S_j = \emptyset$.
- F: for all $1 \leq i \leq n$, $f_i \in F$ is such that $f_i(a) = p_i$ iff $a(x_i) = 1$.

Soft Constraint and Constraint Optimization Equivalence

Soft Constraint Problem $P = (V, D, C, F)$ as a Constraint Optimization Problem

Define $g : \mathcal{A}(V) \rightarrow \mathcal{R}$ so that, given assignment $a \in \mathcal{A}(V)$,

$$g(a) = \sum_{f \in F} f(\pi_{\text{var}(f)}(a)).$$

Then a is an optimal solution to P iff it is an optimal solution to $P_{\text{opt}} = (V, D, C, g)$.

Soft Constraint and Constraint Optimization Equivalence

Soft Constraint Problem $P = (V, D, C, F)$ as a Constraint Optimization Problem

Define $g : \mathcal{A}(V) \to \mathcal{R}$ so that, given assignment $a \in \mathcal{A}(V)$,

$$g(a) = \sum_{f \in F} f(\pi_{\text{var}(f)}(a)).$$

Then a is an optimal solution to P iff it is an optimal solution to $P_{\text{opt}} = (V, D, C, g)$.

Constraint Optimization Problem $P = (V, D, C, f)$ as a Soft Constraint Problem

$P_{\text{soft}} = (V, D, C, F)$, where $F = \{f\}$.

Outline

1. Introduction
2. Branch and Bound
3. Russian Doll Search
4. Dynamic Programming Optimization

Soft Constraint Upper-Bound Functions

Defining $f_{\max} : \mathcal{A}(V) \rightarrow \mathcal{R}$ for Soft Constraint f

$$f_{\max}(a) = \max_{a \sqsubseteq b}(f(b)),$$

where $a \sqsubseteq b$ means that assignment b either equals a, or is an extension of a. $f_{\max}$ is called the **upper-bound function** of f.

Upper-Bound Function Example

$f(x, y, z, w) = 3x - 5y + 10z - 7w$

Upper-Bound Function Example

$f(x, y, z, w) = 3x - 5y + 10z - 7w$

- $\text{dom}(x) = \text{dom}(y) = \text{dom}(z) = \text{dom}(w) = \{0, 1, \ldots, 10\}$

Upper-Bound Function Example

$f(x, y, z, w) = 3x - 5y + 10z - 7w$

- $\text{dom}(x) = \text{dom}(y) = \text{dom}(z) = \text{dom}(w) = \{0, 1, \ldots, 10\}$
- $a = \emptyset$: $f_{\max}(a) = \max_{(x,y,z,w)} (f(x, y, z, w)) = 130$

Upper-Bound Function Example

$f(x, y, z, w) = 3x - 5y + 10z - 7w$

- $\text{dom}(x) = \text{dom}(y) = \text{dom}(z) = \text{dom}(w) = \{0, 1, \ldots, 10\}$
- $a = \emptyset$: $f_{\max}(a) = \max\limits_{(x,y,z,w)}(f(x, y, z, w)) = 130$
- $a = (x = 3, y = 4)$: $f_{\max}(a) = \max\limits_{(z,w)}(10z - 7w - 11) = 89$

Upper-Bound Function Example

$f(x, y, z, w) = 3x - 5y + 10z - 7w$

- $\text{dom}(x) = \text{dom}(y) = \text{dom}(z) = \text{dom}(w) = \{0, 1, \ldots, 10\}$
- $a = \emptyset$: $f_{\max}(a) = \max\limits_{(x,y,z,w)}(f(x, y, z, w)) = 130$
- $a = (x = 3, y = 4)$: $f_{\max}(a) = \max\limits_{(z,w)}(10z - 7w - 11) = 89$
- $a = (x = 7, y = 2, w = 5)$:
  $f_{\max}(a) = \max\limits_{z \in \text{dom}(z)}(10z - 24) = 76$

Upper-Bound Function Example

$f(x, y, z, w) = 3x - 5y + 10z - 7w$

- $\text{dom}(x) = \text{dom}(y) = \text{dom}(z) = \text{dom}(w) = \{0, 1, \ldots, 10\}$
- $a = \emptyset$: $f_{\max}(a) = \max\limits_{(x,y,z,w)} (f(x, y, z, w)) = 130$
- $a = (x = 3, y = 4)$: $f_{\max}(a) = \max\limits_{(z,w)}(10z - 7w - 11) = 89$
- $a = (x = 7, y = 2, w = 5)$:
  $f_{\max}(a) = \max\limits_{z \in \text{dom}(z)} (10z - 24) = 76$
- $a = (7, 2, 6, 5)$: $f_{\max}(a) = f(a) = 14$

Branch and Bound Method for $P = (V, D, C, f)$

Description of Branch and Bound

Branch and Bound Method for $P = (V, D, C, f)$

Description of Branch and Bound

- Extends the tree search framework; assumes use of the forward-checking childbearing rule.

Branch and Bound Method for $P = (V, D, C, f)$

Description of Branch and Bound

- Extends the tree search framework; assumes use of the forward-checking childbearing rule.
- During search, the lower bound L represents the greatest value of f observed thus far.

Branch and Bound Method for $P = (V, D, C, f)$

Description of Branch and Bound

- Extends the tree search framework; assumes use of the forward-checking childbearing rule.
- During search, the lower bound L represents the greatest value of f observed thus far.
- For node/assignment a to bear children, it is necessary that constraint $f_{\max}(a) > L$ be satisfied.

Branch and Bound Method for $P = (V, D, C, f)$

Description of Branch and Bound

- Extends the tree search framework; assumes use of the forward-checking childbearing rule.
- During search, the lower bound L represents the greatest value of f observed thus far.
- For node/assignment a to bear children, it is necessary that constraint $f_{\max}(a) > L$ be satisfied.
- If a solution s is found for which $f(s) > L$, then $L \leftarrow f(s)$, and search continues.

Branch and Bound Example: Combinatorial Bidding

See branch_and_bound_example.pdf

Outline

Defining Subproblems via Projections

The Projection of $P = (V, D, C)$ onto U

Let $U \subset V$ be a nonempty subset of V. Then the **projection of** P **onto** U, denoted $\pi_U(P) = (U, D_U, C_U)$, is defined so that $D_U \subset D$ is the set of domains of variables in U, and $c \in C_U \subseteq C$ iff $\text{var}(c) \subseteq U$. Note: F_U is defined similarly in case P has soft constraint set F.

Defining Subproblems via Projections

The Projection of $P = (V, D, C)$ onto U

Let $U \subset V$ be a nonempty subset of V. Then the **projection of P onto U**, denoted $\pi_U(P) = (U, D_U, C_U)$, is defined so that $D_U \subset D$ is the set of domains of variables in U, and $c \in C_U \subseteq C$ iff $\text{var}(c) \subseteq U$. Note: F_U is defined similarly in case P has soft constraint set F.

Projection Example

Defining Subproblems via Projections

The Projection of $P = (V, D, C)$ onto U

Let $U \subset V$ be a nonempty subset of V. Then the **projection of** P **onto** U, denoted $\pi_U(P) = (U, D_U, C_U)$, is defined so that $D_U \subset D$ is the set of domains of variables in U, and $c \in C_U \subseteq C$ iff $\text{var}(c) \subseteq U$. Note: F_U is defined similarly in case P has soft constraint set F.

Projection Example

- $P = (V, D, C, F)$, where $V = \{x, y, z, w, t\}$, $C = \{c_1, c_2, c_3\}$, $F = \{f_1, f_2\}$, $\text{var}(c_1) = \{x, y, z\}$, $\text{var}(c_2) = \{y, w\}$, $\text{var}(c_3) = \{w, t\}$, $\text{var}(f_1) = \{y, t\}$, $\text{var}(f_2) = \{y, z, w\}$, $U = \{y, w, t\}$.

Defining Subproblems via Projections

The Projection of $P = (V, D, C)$ onto U

Let $U \subset V$ be a nonempty subset of V. Then the **projection of** P **onto** U, denoted $\pi_U(P) = (U, D_U, C_U)$, is defined so that $D_U \subset D$ is the set of domains of variables in U, and $c \in C_U \subseteq C$ iff $\text{var}(c) \subseteq U$. Note: F_U is defined similarly in case P has soft constraint set F.

Projection Example

- $P = (V, D, C, F)$, where $V = \{x, y, z, w, t\}$, $C = \{c_1, c_2, c_3\}$, $F = \{f_1, f_2\}$, $\text{var}(c_1) = \{x, y, z\}$, $\text{var}(c_2) = \{y, w\}$, $\text{var}(c_3) = \{w, t\}$, $\text{var}(f_1) = \{y, t\}$, $\text{var}(f_2) = \{y, z, w\}$, $U = \{y, w, t\}$.
- $\pi_U(P) = (U, D_U, C_U, F_U)$, where $D_U = \{\text{dom}(y), \text{dom}(w), \text{dom}(t)\}$, $C_U = \{c_2, c_3\}$, and $F_U = \{f_1\}$.

Russian Doll Search

Russian Doll Search

Overview

Russian doll search solves increasingly larger optimization subproblems via branch and bound. The solutions to the smaller subproblems are used to obtain both lower bounds and tighter upper bounds on the optimal objective values for the larger subproblems.

Russian Doll Search Algorithm

Subproblems of $P = (\{x_1, \ldots, x_n\}, D, C, C_s)$

Russian Doll Search Algorithm

Subproblems of $P = (\{x_1, \ldots, x_n\}, D, C, C_s)$

- **Subproblem P_n.** $U_n = \{x_n\}$. $P_n = \pi_{U_n}(P)$. M_n: optimal objective value for P_n.

Russian Doll Search Algorithm

Subproblems of $P = (\{x_1, \ldots, x_n\}, D, C, C_s)$

- **Subproblem P_n.** $U_n = \{x_n\}$. $P_n = \pi_{U_n}(P)$. M_n: optimal objective value for P_n.
- **Subproblem P_{n-i}, $i = 1, \ldots, n-1$.** $U_{n-i} = \{x_{n-i}, \ldots, x_n\}$, $P_{n-i} = \pi_{U_{n-i}}(P)$.

Russian Doll Search Algorithm

Subproblems of $P = (\{x_1, \ldots, x_n\}, D, C, C_s)$

- **Subproblem P_n.** $U_n = \{x_n\}$. $P_n = \pi_{U_n}(P)$. M_n: optimal objective value for P_n.
- **Subproblem P_{n-i}, $i = 1, \ldots, n-1$.** $U_{n-i} = \{x_{n-i}, \ldots, x_n\}$, $P_{n-i} = \pi_{U_{n-i}}(P)$.

Computing Initial Lower Bound L for P_{n-i}

If $i = 0$, then $L = -\infty$. Otherwise, let b be an optimal solution for P_{n-i+1}. Then

$$L = M_{n-i+1} + \max_e \Big(\sum_f f(\pi_{\mathrm{var}(f)}(e))\Big),$$

where the max is taken over all extensions e of b over U_{n-i} that satisfy P_{n-i}; while sum is taken over all $f \in F_{U_{n-i}}$ for which $x_{n-i} \in \mathrm{var}(f)$.

Russian Doll Search Algorithm

Computing Upper Bound U for Assignment a in Problem P_{n-i}

Let a be an assignment over $W = \{x_{n-i}, \ldots, x_{n-j}\}$ that is consistent with P_{n-i}. Partition $F_{U_{n-i}}$ into three sets: X, Y, and Z, where, for $f \in F_{U_{n-i}}$,

Russian Doll Search Algorithm

Computing Upper Bound U for Assignment a in Problem P_{n-i}

Let a be an assignment over $W = \{x_{n-i}, \ldots, x_{n-j}\}$ that is consistent with P_{n-i}. Partition $F_{U_{n-i}}$ into three sets: X, Y, and Z, where, for $f \in F_{U_{n-i}}$,

- $f \in X$ iff $\text{var}(f) \subseteq W$ (functions with *all* variables assigned by a),

Russian Doll Search Algorithm

Computing Upper Bound U for Assignment a in Problem P_{n-i}

Let a be an assignment over $W = \{x_{n-i}, \ldots, x_{n-j}\}$ that is consistent with P_{n-i}. Partition $F_{U_{n-i}}$ into three sets: X, Y, and Z, where, for $f \in F_{U_{n-i}}$,

- $f \in X$ iff $\text{var}(f) \subseteq W$ (functions with *all* variables assigned by a),
- $f \in Y$ iff $\text{var}(f) \subseteq U_{n-j+1}$ (functions with *no* variables assigned by a),

Russian Doll Search Algorithm

Computing Upper Bound U for Assignment a in Problem P_{n-i}

Let a be an assignment over $W = \{x_{n-i}, \ldots, x_{n-j}\}$ that is consistent with P_{n-i}. Partition $F_{U_{n-i}}$ into three sets: X, Y, and Z, where, for $f \in F_{U_{n-i}}$,

- $f \in X$ iff $\text{var}(f) \subseteq W$ (functions with *all* variables assigned by a),
- $f \in Y$ iff $\text{var}(f) \subseteq U_{n-j+1}$ (functions with *no* variables assigned by a),
- $f \in Z$ iff f is neither in X nor in Y (functions with *some* variables assigned by a),

Russian Doll Search Algorithm

Computing Upper Bound U for Assignment a in Problem P_{n-i}

Let a be an assignment over $W = \{x_{n-i}, \ldots, x_{n-j}\}$ that is consistent with P_{n-i}. Partition $F_{U_{n-i}}$ into three sets: X, Y, and Z, where, for $f \in F_{U_{n-i}}$,

- $f \in X$ iff $\text{var}(f) \subseteq W$ (functions with *all* variables assigned by a),
- $f \in Y$ iff $\text{var}(f) \subseteq U_{n-j+1}$ (functions with *no* variables assigned by a),
- $f \in Z$ iff f is neither in X nor in Y (functions with *some* variables assigned by a),

Then $U = \sum_{f \in X} f(a) + \sum_{f \in Z} f_{\max}(a) + M_{n-j+1}$

Russian Doll Search Example: Combinatorial Bidding

Subproblems $P_8, \ldots, P_2$

Russian Doll Search Example: Combinatorial Bidding

Subproblems $P_8, \ldots, P_2$

- P_8. Bids: $B_8 = \{(\{1,4\},4)\}$. $M_8 = 4$.

Russian Doll Search Example: Combinatorial Bidding

Subproblems $P_8, \ldots, P_2$

- P_8. Bids: $B_8 = \{(\{1,4\},4)\}$. $M_8 = 4$.
- P_7. Bids: $B_7 = B_8 + \{(\{2,4,5,7\},6)\}$. $M_7 = 6$.

Russian Doll Search Example: Combinatorial Bidding

Subproblems $P_8, \ldots, P_2$

- P_8. Bids: $B_8 = \{(\{1,4\},4)\}$. $M_8 = 4$.
- P_7. Bids: $B_7 = B_8 + \{(\{2,4,5,7\},6)\}$. $M_7 = 6$.
- P_6. Bids: $B_6 = B_7 + \{(\{3,4,5\},7)\}$. $M_6 = 7$.

Russian Doll Search Example: Combinatorial Bidding

Subproblems $P_8, \ldots, P_2$

- P_8. Bids: $B_8 = \{(\{1,4\},4)\}$. $M_8 = 4$.
- P_7. Bids: $B_7 = B_8 + \{(\{2,4,5,7\},6)\}$. $M_7 = 6$.
- P_6. Bids: $B_6 = B_7 + \{(\{3,4,5\},7)\}$. $M_6 = 7$.
- P_5. Bids: $B_5 = B_6 + \{(\{1,2,4,5,6\},8)\}$. $M_5 = 8$.

Russian Doll Search Example: Combinatorial Bidding

Subproblems $P_8, \ldots, P_2$

- P_8. Bids: $B_8 = \{(\{1,4\},4)\}$. $M_8 = 4$.
- P_7. Bids: $B_7 = B_8 + \{(\{2,4,5,7\},6)\}$. $M_7 = 6$.
- P_6. Bids: $B_6 = B_7 + \{(\{3,4,5\},7)\}$. $M_6 = 7$.
- P_5. Bids: $B_5 = B_6 + \{(\{1,2,4,5,6\},8)\}$. $M_5 = 8$.
- P_4. Bids: $B_4 = B_5 + \{(\{1,8\},5)\}$. $M_4 = 12$.

Russian Doll Search Example: Combinatorial Bidding

Subproblems $P_8, \ldots, P_2$

- P_8. Bids: $B_8 = \{(\{1,4\},4)\}$. $M_8 = 4$.
- P_7. Bids: $B_7 = B_8 + \{(\{2,4,5,7\},6)\}$. $M_7 = 6$.
- P_6. Bids: $B_6 = B_7 + \{(\{3,4,5\},7)\}$. $M_6 = 7$.
- P_5. Bids: $B_5 = B_6 + \{(\{1,2,4,5,6\},8)\}$. $M_5 = 8$.
- P_4. Bids: $B_4 = B_5 + \{(\{1,8\},5)\}$. $M_4 = 12$.
- P_3. Bids: $B_3 = B_4 + \{(\{1,5,8\},6)\}$. $M_3 = 12$.

Russian Doll Search Example: Combinatorial Bidding

Subproblems $P_8, \ldots, P_2$

- P_8. Bids: $B_8 = \{(\{1,4\},4)\}$. $M_8 = 4$.
- P_7. Bids: $B_7 = B_8 + \{(\{2,4,5,7\},6)\}$. $M_7 = 6$.
- P_6. Bids: $B_6 = B_7 + \{(\{3,4,5\},7)\}$. $M_6 = 7$.
- P_5. Bids: $B_5 = B_6 + \{(\{1,2,4,5,6\},8)\}$. $M_5 = 8$.
- P_4. Bids: $B_4 = B_5 + \{(\{1,8\},5)\}$. $M_4 = 12$.
- P_3. Bids: $B_3 = B_4 + \{(\{1,5,8\},6)\}$. $M_3 = 12$.
- P_2. Bids: $B_2 = B_3 + \{(\{7,8\},4)\}$. $M_2 = 12$.

Russian Doll Example: Combinatorial Bidding

Assignment	Items	Price	L	U
x_1	$\{2, 3, 5, 7, 8\}$	7	12	$19 = 7 + M_2$
x_2	$\{7, 8\}$	4	12	
x_3	$\{1, 5, 8\}$	6	12	
x_4	$\{1, 8\}$	5	12	
x_5	$\{1, 2, 4, 5, 6\}$	8	12	
x_6	$\{3, 4, 5\}$	7	12	
x_7	$\{2, 4, 5, 7\}$	6	12	
x_8	$\{1, 4\}$	4	12	

Figure: $L = 12$ via P_2 solution extension $(0, 0, 0, 1, 0, 1, 0, 0)$

Russian Doll Example: Combinatorial Bidding

Assignment	Items	Price	L	U
$x_1 = 1$	$\{2,3,5,7,8\}$	7	12	$19 = 7 + M_2$
x_2	$\{7,8\}$	4	12	
x_3	$\{1,5,8\}$	6	12	
x_4	$\{1,8\}$	5	12	
x_5	$\{1,2,4,5,6\}$	8	12	
x_6	$\{3,4,5\}$	7	12	
x_7	$\{2,4,5,7\}$	6	12	
x_8	$\{1,4\}$	4	12	

Russian Doll Example: Combinatorial Bidding

Assignment	Items	Price	L	U
$x_1 = 1$	$\{2, 3, 5, 7, 8\}$	7	12	$19 = 7 + M_2$
$x_2 = 1$	$\{7, 8\}$	4	12	
x_3	$\{1, 5, 8\}$	6	12	
x_4	$\{1, 8\}$	5	12	
x_5	$\{1, 2, 4, 5, 6\}$	8	12	
x_6	$\{3, 4, 5\}$	7	12	
x_7	$\{2, 4, 5, 7\}$	6	12	
x_8	$\{1, 4\}$	4	12	

Russian Doll Example: Combinatorial Bidding

Assignment	Items	Price	L	U
$x_1 = 1$	$\{2,3,5,7,8\}$	7	12	$19 = 7 + M_2$
$x_2 = 0$	$\{7,8\}$	4	12	$19 = 7 + M_3$
x_3	$\{1,5,8\}$	6	12	
x_4	$\{1,8\}$	5	12	
x_5	$\{1,2,4,5,6\}$	8	12	
x_6	$\{3,4,5\}$	7	12	
x_7	$\{2,4,5,7\}$	6	12	
x_8	$\{1,4\}$	4	12	

Russian Doll Example: Combinatorial Bidding

Assignment	Items	Price	L	U
$x_1 = 1$	$\{2,3,5,7,8\}$	7	12	$19 = 7 + M_2$
$x_2 = 0$	$\{7,8\}$	4	12	$19 = 7 + M_3$
$x_3 = 1$	$\{1,5,8\}$	6	12	
x_4	$\{1,8\}$	5	12	
x_5	$\{1,2,4,5,6\}$	8	12	
x_6	$\{3,4,5\}$	7	12	
x_7	$\{2,4,5,7\}$	6	12	
x_8	$\{1,4\}$	4	12	

Russian Doll Example: Combinatorial Bidding

Assignment	Items	Price	L	U
$x_1 = 1$	$\{2,3,5,7,8\}$	7	12	$19 = 7 + M_2$
$x_2 = 0$	$\{7,8\}$	4	12	$19 = 7 + M_3$
$x_3 = 0$	$\{1,5,8\}$	6	12	$19 = 7 + M_4$
x_4	$\{1,8\}$	5	12	
x_5	$\{1,2,4,5,6\}$	8	12	
x_6	$\{3,4,5\}$	7	12	
x_7	$\{2,4,5,7\}$	6	12	
x_8	$\{1,4\}$	4	12	

Russian Doll Example: Combinatorial Bidding

Assignment	Items	Price	L	U
$x_1 = 1$	$\{2,3,5,7,8\}$	7	12	$19 = 7 + M_2$
$x_2 = 0$	$\{7,8\}$	4	12	$19 = 7 + M_3$
$x_3 = 0$	$\{1,5,8\}$	6	12	$19 = 7 + M_4$
$x_4 = 1$	$\{1,8\}$	5	12	
x_5	$\{1,2,4,5,6\}$	8	12	
x_6	$\{3,4,5\}$	7	12	
x_7	$\{2,4,5,7\}$	6	12	
x_8	$\{1,4\}$	4	12	

Russian Doll Example: Combinatorial Bidding

Assignment	Items	Price	L	U
$x_1 = 1$	$\{2, 3, 5, 7, 8\}$	7	12	$19 = 7 + M_2$
$x_2 = 0$	$\{7, 8\}$	4	12	$19 = 7 + M_3$
$x_3 = 0$	$\{1, 5, 8\}$	6	12	$19 = 7 + M_4$
$x_4 = 0$	$\{1, 8\}$	5	12	$15 = 7 + M_5$
x_5	$\{1, 2, 4, 5, 6\}$	8	12	
x_6	$\{3, 4, 5\}$	7	12	
x_7	$\{2, 4, 5, 7\}$	6	12	
x_8	$\{1, 4\}$	4	12	

Russian Doll Example: Combinatorial Bidding

Assignment	Items	Price	L	U
$x_1 = 1$	$\{2, 3, 5, 7, 8\}$	7	12	$19 = 7 + M_2$
$x_2 = 0$	$\{7, 8\}$	4	12	$19 = 7 + M_3$
$x_3 = 0$	$\{1, 5, 8\}$	6	12	$19 = 7 + M_4$
$x_4 = 0$	$\{1, 8\}$	5	12	$15 = 7 + M_5$
$x_5 = 1$	$\{1, 2, 4, 5, 6\}$	8	12	
x_6	$\{3, 4, 5\}$	7	12	
x_7	$\{2, 4, 5, 7\}$	6	12	
x_8	$\{1, 4\}$	4	12	

Russian Doll Example: Combinatorial Bidding

Assignment	Items	Price	L	U
$x_1 = 1$	$\{2,3,5,7,8\}$	7	12	$19 = 7 + M_2$
$x_2 = 0$	$\{7,8\}$	4	12	$19 = 7 + M_3$
$x_3 = 0$	$\{1,5,8\}$	6	12	$19 = 7 + M_4$
$x_4 = 0$	$\{1,8\}$	5	12	$15 = 7 + M_5$
$x_5 = 0$	$\{1,2,4,5,6\}$	8	12	$14 = 7 + M_6$
x_6	$\{3,4,5\}$	7	12	
x_7	$\{2,4,5,7\}$	6	12	
x_8	$\{1,4\}$	4	12	

Russian Doll Example: Combinatorial Bidding

Assignment	Items	Price	L	U
$x_1 = 1$	$\{2,3,5,7,8\}$	7	12	$19 = 7 + M_2$
$x_2 = 0$	$\{7,8\}$	4	12	$19 = 7 + M_3$
$x_3 = 0$	$\{1,5,8\}$	6	12	$19 = 7 + M_4$
$x_4 = 0$	$\{1,8\}$	5	12	$15 = 7 + M_5$
$x_5 = 0$	$\{1,2,4,5,6\}$	8	12	$14 = 7 + M_6$
$x_6 = 1$	$\{3,4,5\}$	7	12	
x_7	$\{2,4,5,7\}$	6	12	
x_8	$\{1,4\}$	4	12	

Russian Doll Example: Combinatorial Bidding

Assignment	Items	Price	L	U
$x_1 = 1$	$\{2,3,5,7,8\}$	7	12	$19 = 7 + M_2$
$x_2 = 0$	$\{7,8\}$	4	12	$19 = 7 + M_3$
$x_3 = 0$	$\{1,5,8\}$	6	12	$19 = 7 + M_4$
$x_4 = 0$	$\{1,8\}$	5	12	$15 = 7 + M_5$
$x_5 = 0$	$\{1,2,4,5,6\}$	8	12	$14 = 7 + M_6$
$x_6 = 0$	$\{3,4,5\}$	7	12	$13 = 7 + M_6$
x_7	$\{2,4,5,7\}$	6	12	
x_8	$\{1,4\}$	4	12	

Russian Doll Example: Combinatorial Bidding

Assignment	Items	Price	L	U
$x_1 = 1$	$\{2,3,5,7,8\}$	7	12	$19 = 7 + M_2$
$x_2 = 0$	$\{7,8\}$	4	12	$19 = 7 + M_3$
$x_3 = 0$	$\{1,5,8\}$	6	12	$19 = 7 + M_4$
$x_4 = 0$	$\{1,8\}$	5	12	$15 = 7 + M_5$
$x_5 = 0$	$\{1,2,4,5,6\}$	8	12	$14 = 7 + M_6$
$x_6 = 0$	$\{3,4,5\}$	7	12	$13 = 7 + M_7$
$x_7 = 1$	$\{2,4,5,7\}$	6	12	
x_8	$\{1,4\}$	4	12	

Russian Doll Example: Combinatorial Bidding

Assignment	Items	Price	L	U
$x_1 = 1$	$\{2, 3, 5, 7, 8\}$	7	12	$19 = 7 + M_2$
$x_2 = 0$	$\{7, 8\}$	4	12	$19 = 7 + M_3$
$x_3 = 0$	$\{1, 5, 8\}$	6	12	$19 = 7 + M_4$
$x_4 = 0$	$\{1, 8\}$	5	12	$15 = 7 + M_5$
$x_5 = 0$	$\{1, 2, 4, 5, 6\}$	8	12	$14 = 7 + M_6$
$x_6 = 0$	$\{3, 4, 5\}$	7	12	$13 = 7 + M_6$
$x_7 = 0$	$\{2, 4, 5, 7\}$	6	12	$11 = 7 + M_8$
x_8	$\{1, 4\}$	4	12	

Russian Doll Example: Combinatorial Bidding

Assignment	Items	Price	Lower Bound	U
$x_1 = 0$	$\{2,3,5,7,8\}$	7	12	$12 = M_2$
x_2	$\{7,8\}$	4	12	
x_3	$\{1,5,8\}$	6	12	
x_4	$\{1,8\}$	5	12	
x_5	$\{1,2,4,5,6\}$	8	12	
x_6	$\{3,4,5\}$	7	12	
x_7	$\{2,4,5,7\}$	6	12	
x_8	$\{1,4\}$	4	12	

Russian Doll Example: Combinatorial Bidding

Assignment	Items	Price	Lower Bound	U
x_1	$\{2, 3, 5, 7, 8\}$	7	12	
x_2	$\{7, 8\}$	4	12	
x_3	$\{1, 5, 8\}$	6	12	
x_4	$\{1, 8\}$	5	12	
x_5	$\{1, 2, 4, 5, 6\}$	8	12	
x_6	$\{3, 4, 5\}$	7	12	
x_7	$\{2, 4, 5, 7\}$	6	12	
x_8	$\{1, 4\}$	4	12	

Figure: Optimal solution: $(0, 0, 0, 1, 0, 1, 0, 0)$ with optimal objective value 12.

Russian Doll Search Example: Table Functions

Objective: maximize $\sum_{i=1}^{6} f_i$

x_4	f_1
0	4
1	6

x_3	x_4	f_2
0	0	0
0	1	6
1	0	4
1	1	6

x_2	x_3	f_3
0	0	3
0	1	3
1	0	0
1	1	8
2	0	2
2	1	1

x_2	x_4	f_4
0	0	3
0	1	2
1	0	1
1	1	5
2	0	1
2	1	3

Russian Doll Search Example: Table Functions

Objective: maximize $\sum_{i=1}^{6} f_i$

x_1	x_2	x_3	f_5
0	0	0	3
0	0	1	6
0	1	0	6
0	1	1	5
0	2	0	4
0	2	1	3
1	0	0	2
1	0	1	1
1	1	0	3
1	1	1	5
1	2	0	4
1	2	1	2
2	X	X	2

x_1	x_3	f_6
0	0	5
0	1	4
1	0	5
1	1	6
2	0	4
2	1	2

Russian Doll Search Example: Table Functions

Solve P_4

Assignment	U	L
x_4	$f_{1,\max}(x_4) = 6$	$-\infty$

Russian Doll Search Example: Table Functions

Solve P_4

Assignment	U	L
$x_4 = 0$	$f_1(0) = 4$	4

Russian Doll Search Example: Table Functions

Solve P_4

Assignment	U	L
$x_4 = 1$	$f_1(1) = 6$	6

Russian Doll Search Example: Table Functions

Solve P_4

Assignment	U	L
x_4		6

Figure: Optimal solution: $x_4 = 1$, Optimal objective value: $M_4 = 6$.

Russian Doll Search Example: Table Functions

Solve P_3

Assignment	U	L
x_3	$f_{2,\max}(x_3, x_4) + M_4 = 12$	12
x_4		12

Figure: $L = 12$ via P_4 solution extension $(0, 1)$

Russian Doll Search Example: Table Functions

Solve P_3

Assignment	U	L
$x_3 = 0$	$f_{2,\max}(0, x_4) + M_4 = 12$	12
x_4		12

Russian Doll Search Example: Table Functions

Solve P_3

Assignment	U	L
$x_3 = 1$	$f_{2,\max}(1, x_4) + M_4 = 12$	12
x_4		12

Russian Doll Search Example: Table Functions

Solve P_3

Assignment	U	L
x_3		12
x_4		12

Figure: Optimal solution: $(0, 1)$, Optimal objective value: $M_3 = 12$.

Russian Doll Search Example: Table Functions

Solve P_2

Assignment	U	L
x_2	$f_{3,\max} + f_{4,\max} + M_3 = 25$	17
x_3		17
x_4		17

Figure: $L = 17$ via P_3 solution extension $(0, 0, 1)$

Russian Doll Search Example: Table Functions

Solve P_2

Assignment	U	L
$x_2 = 0$	$f_{3,\max} + f_{4,\max} + M_3 = 18$	17
x_3		17
x_4		17

Russian Doll Search Example: Table Functions

Solve P_2

Assignment	U	L
$x_2 = 0$	$f_{3,\max} + f_{4,\max} + M_3 = 18$	17
$x_3 = 0$	$f_3 + f_{2,\max} + f_{4,\max} + M_4 = 18$	17
x_4		17

Russian Doll Search Example: Table Functions

Solve P_2

Assignment	U	L
$x_2 = 0$	$f_{3,\text{max}} + f_{4,\text{max}} + M_3 = 18$	17
$x_3 = 0$	$f_3 + f_{2,\text{max}} + f_{4,\text{max}} + M_4 = 18$	17
$x_4 = 0$	$f_1 + f_2 + f_3 + f_4 = 10$	17

Russian Doll Search Example: Table Functions

Solve P_2

Assignment	U	L
$x_2 = 0$	$f_{3,\max} + f_{4,\max} + M_3 = 18$	17
$x_3 = 0$	$f_3 + f_{2,\max} + f_{4,\max} + M_4 = 18$	17
$x_4 = 1$	$f_1 + f_2 + f_3 + f_4 = 17$	17

Russian Doll Search Example: Table Functions

Solve P_2

Assignment	U	L
$x_2 = 0$	$f_{3,\max} + f_{4,\max} + M_3 = 18$	17
$x_3 = 1$	$f_3 + f_{2,\max} + f_{4,\max} + M_4 = 17$	17
x_4		17

Russian Doll Search Example: Table Functions

Solve P_2

Assignment	U	L
$x_2 = 1$	$f_{3,\max} + f_{4,\max} + M_3 = 25$	17
x_3		17
x_4		17

Russian Doll Search Example: Table Functions

Solve P_2

Assignment	U	L
$x_2 = 1$	$f_{3,\max} + f_{4,\max} + M_3 = 25$	17
$x_3 = 0$	$f_3 + f_{2,\max} + f_{4,\max} + M_4 = 17$	17
x_4		17

Russian Doll Search Example: Table Functions

Solve P_2

Assignment	U	L
$x_2 = 1$	$f_{3,\max} + f_{4,\max} + M_3 = 25$	17
$x_3 = 1$	$f_3 + f_{2,\max} + f_{4,\max} + M_4 = 25$	17
x_4		17

Russian Doll Search Example: Table Functions

Solve P_2

Assignment	U	L
$x_2 = 1$	$f_{3,\text{max}} + f_{4,\text{max}} + M_3 = 25$	21
$x_3 = 1$	$f_3 + f_{2,\text{max}} + f_{4,\text{max}} + M_4 = 25$	21
$x_4 = 0$	$f_1 + f_2 + f_3 + f_4 = 21$	21

Russian Doll Search Example: Table Functions

Solve P_2

Assignment	U	L
$x_2 = 1$	$f_{3,\max} + f_{4,\max} + M_3 = 25$	25
$x_3 = 1$	$f_3 + f_{2,\max} + f_{4,\max} + M_4 = 25$	25
$x_4 = 1$	$f_1 + f_2 + f_3 + f_4 = 25$	25

Russian Doll Search Example: Table Functions

Solve P_2

Assignment	U	L
$x_2 = 2$	$f_{3,\max} + f_{4,\max} + M_3 = 17$	25
x_3		25
x_4		25

Russian Doll Search Example: Table Functions

Solve P_2

Assignment	U	L
x_2		25
x_3		25
x_4		25

Figure: Optimal solution: $(1, 1, 1)$, Optimal objective value: $M_2 = 25$.

Russian Doll Search Example: Table Functions

Solve P_1

Assignment	U	L
x_1	$f_{5,\max} + f_{6,\max} + M_2 = 37$	36
x_2		36
x_3		36
x_4		36

Figure: $L = 36$ via P_2 solution extension $(1, 1, 1, 1)$

Russian Doll Search Example: Table Functions

Solve P_1

Assignment	U	L
$x_1 = 0$	$f_{5,\max} + f_{6,\max} + M_2 = 36$	36
x_2		36
x_3		36
x_4		36

Russian Doll Search Example: Table Functions

Solve P_1

Assignment	U	L
$x_1 = 1$	$f_{5,\max} + f_{6,\max} + M_2 = 36$	36
x_2		36
x_3		36
x_4		36

Russian Doll Search Example: Table Functions

Solve P_1

Assignment	U	L
$x_1 = 2$	$f_{5,\max} + f_{6,\max} + M_2 = 31$	36
x_2		36
x_3		36
x_4		36

Russian Doll Search Example: Table Functions

Solve P_1

Assignment	U	L
x_1		36
x_2		36
x_3		36
x_4		36

Figure: Optimal solution: $(1, 1, 1, 1)$, Optimal objective value: $M_1 = 36$.

Outline

Dynamic Programming Approach to Function Optimization

A Bottom-up Approach to Optimization

Dynamic Programming Approach to Function Optimization

A Bottom-up Approach to Optimization

- **Goal.** Find an assignment a over $\{x_1, \ldots, x_n\}$ that maximizes

$$F(x_1, \ldots, x_n) = \sum_{i=1}^{k} f_i.$$

Dynamic Programming Approach to Function Optimization

A Bottom-up Approach to Optimization

- **Goal.** Find an assignment a over $\{x_1, \ldots, x_n\}$ that maximizes

$$F(x_1, \ldots, x_n) = \sum_{i=1}^{k} f_i.$$

- **Strategy.** Optimize sub-problems that depend on fewer variables and fewer functions (f_i's). Use these sub-problem solutions to optimize over increasingly larger sub-problems, up to and including $F(x_1, \ldots, x_n)$.

Dynamic Programming Approach to Function Optimization

A Bottom-up Approach to Optimization

- **Goal.** Find an assignment a over $\{x_1, \ldots, x_n\}$ that maximizes

$$F(x_1, \ldots, x_n) = \sum_{i=1}^{k} f_i.$$

- **Strategy.** Optimize sub-problems that depend on fewer variables and fewer functions (f_i's). Use these sub-problem solutions to optimize over increasingly larger sub-problems, up to and including $F(x_1, \ldots, x_n)$.
- **Strategy Execution.** Partition f_i's into **buckets** $B(x_1), \ldots, B(x_n)$. Process the buckets in reverse order; i.e. $B(x_{n-1}), \ldots, B(x_1)$. Each processed bucket corresponds with optimizing over a sub-problem. This approach is sometimes called **variable elimination**, or **bucket elimination**.

Bucket Elimination Example

Goal: maximize

$f_1(u) + f_2(u, x) + f_3(u, w, y) + f_4(u, y) + f_5(v, x, y) + f_6(v, z)$

Assume variable ordering u, v, w, x, y, z. Then

$$\max_{u,v,w,x,y,z} f_1(u)+f_2(u, x)+f_3(u, w, y)+f_4(u, y)+f_5(v, x, y)+f_6(v, z) =$$

Bucket Elimination Example

Goal: maximize

$f_1(u) + f_2(u,x) + f_3(u,w,y) + f_4(u,y) + f_5(v,x,y) + f_6(v,z)$

Assume variable ordering u, v, w, x, y, z. Then

$$\max_{u,v,w,x,y,z} f_1(u)+f_2(u,x)+f_3(u,w,y)+f_4(u,y)+f_5(v,x,y)+f_6(v,z) =$$

$$\max_u f_1(u)+\max_v \max_w \max_x f_2(u,x)+\max_y f_3(u,w,y)+f_4(u,y)+f_5(v,x,y)$$

$$+\max_z f_6(v,z).$$

Bucket Elimination Example

Key Observations

Bucket Elimination Example

Key Observations

- The max operators are distributed from left to right, in accordance with the pre-defined variable ordering.

Bucket Elimination Example

Key Observations

- The max operators are distributed from left to right, in accordance with the pre-defined variable ordering.
- If f is positioned to the left of a $\max_{t}$ operator, for some variable t, then f neither depends on t, nor on any variable to the right of t.

Bucket Elimination Example

Key Observations

- The max operators are distributed from left to right, in accordance with the pre-defined variable ordering.
- If f is positioned to the left of a $\max_t$ operator, for some variable t, then f neither depends on t, nor on any variable to the right of t.
- If $\max_t$ is the first max operator to the left of f, then t is the latest (in terms of the variable ordering) variable for whhich f depends. Moreover, t is the unique variable for which $f \in B(t)$, the bucket of t.

Bucket Elimination Example

$$\max_u f_1(u) + \max_v \max_w \max_x f_2(u,x) + \max_y f_3(u,w,y) + f_4(u,y) + f_5(v,x,y) + \max_z f_6(v,z).$$

Bucket Elimination Example

$$\max_u f_1(u) + \max_v \max_w \max_x f_2(u,x) + \max_y f_3(u,w,y) + f_4(u,y) + f_5(v,x,y)$$

$$+ \max_z f_6(v,z).$$

Pre-processed Buckets

$B(u) = \{f_1(u)\}$, $B(v) = B(w) = \emptyset$, $B(x) = \{f_2(u,x)\}$,
$B(y) = \{f_3(u,w,y), f_4(u,y), f_5(v,x,y)\}$, $B(z) = f_6(v,z)$.

Bucket Elimination Example

Bucket Processing

Bucket Elimination Example

Bucket Processing

- Buckets are processed in reverse variable order.

Bucket Elimination Example

Bucket Processing

- Buckets are processed in reverse variable order.
- For bucket $B(t)$, let $x_1, \ldots, x_n$ denote the variables (other than t) that the functions in $B(t)$ depend on. Then **processing** $B(t)$ means computing the function

$$g_t(x_1, \ldots, x_n) = \max_t F(x_1, \ldots, x_n, t),$$

and placing it in bucket $B(x_n)$ (assuming x_n is the rightmost variable of $x_1, \ldots, x_n$), where $F(x_1, \ldots, x_n, t)$ is the sum of all functions in $B(t)$.

Bucket Elimination Example

Bucket Processing

- Buckets are processed in reverse variable order.
- For bucket $B(t)$, let $x_1, \ldots, x_n$ denote the variables (other than t) that the functions in $B(t)$ depend on. Then **processing** $B(t)$ means computing the function

$$g_t(x_1, \ldots, x_n) = \max_t F(x_1, \ldots, x_n, t),$$

  and placing it in bucket $B(x_n)$ (assuming x_n is the rightmost variable of $x_1, \ldots, x_n$), where $F(x_1, \ldots, x_n, t)$ is the sum of all functions in $B(t)$.
- BS_0 denotes the pre-processed set of buckets. BS_k denotes the state of the buckets after the k latest buckets have been processed. Processed buckets are no longer listed as part of BS_k.

Bucket Elimination Example

BS_0

$B(u) = \{f_1(u)\}$, $B(v) = B(w) = \emptyset$, $B(x) = \{f_2(u, x)\}$,
$B(y) = \{f_3(u, w, y), f_4(u, y), f_5(v, x, y)\}$, $B(z) = f_6(v, z)$.

Process $B(z)$

Compute $g_z(v) = \max_z f_6(v, z)$. Place in $B(v)$.

Bucket Elimination Example

BS_1

$B(u) = \{f_1(u)\}$, $B(v) = \{g_z(v)\}$, $B(w) = \emptyset$, $B(x) = \{f_2(u, x)\}$, $B(y) = \{f_3(u, w, y), f_4(u, y), f_5(v, x, y)\}$

Process $B(y)$

Compute $g_y(u, v, w, x) = \max_y f_3(u, w, y) + f_4(u, y) + f_5(v, x, y)$.

Place in $B(x)$.

Bucket Elimination Example

BS_2

$B(u) = \{f_1(u)\}$, $B(v) = \{g_z(v)\}$, $B(w) = \emptyset$,
$B(x) = \{f_2(u,x), g_y(u,v,w,x)\}$

Process $B(x)$

Compute $g_x(u,v,w) = \max_x f_2(u,x) + g_y(u,v,w,x)$. Place in $B(w)$.

Bucket Elimination Example

BS_3

$B(u) = \{f_1(u)\}$, $B(v) = \{g_z(v)\}$, $B(w) = \{g_x(u, v, w)\}$

Process $B(w)$

Compute $g_w(u, v) = \max_w g_x(u, v, w)$. Place in $B(v)$.

Bucket Elimination Example

BS_4

$B(u) = \{f_1(u)\}$, $B(v) = \{g_z(v), g_w(u, v)\}$

Process $B(v)$

Compute $g_v(u) = \max_v g_z(v) + g_w(u, v)$. Place in $B(u)$.

Bucket Elimination Example

BS_5

$B(u) = \{f_1(u), g_v(u)\}$

Process $B(u)$

Compute $g_u() = \max_u f_1(u) + g_v(u)$. $g_u()$ denotes the maximum value of the original sum!

Bucket Elimination Example: Table Functions

Variables and Functions

Bucket Elimination Example: Table Functions

Variables and Functions

- **Variable ordering:** x_1, x_2, x_3, x_4.

Bucket Elimination Example: Table Functions

Variables and Functions

- **Variable ordering:** x_1, x_2, x_3, x_4.
- **Functions:** $f_1(x_4)$, $f_2(x_3, x_4)$, $f_3(x_2, x_3)$, $f_4(x_2, x_4)$, $f_5(x_1, x_2, x_3)$, $f_6(x_1, x_3)$

Bucket Elimination Example: Table Functions

Process bucket $B(x_4) = \{f_1(x_4), f_2(x_3, x_4), f_4(x_2, x_4)\}$

x_4	f_1
0	4
1	6

x_3	x_4	f_2
0	0	0
0	1	6
1	0	4
1	1	6

x_2	x_4	f_4
0	0	3
0	1	2
1	0	1
1	1	5
2	0	1
2	1	3

x_2	x_3	$g_{x_4}(x_2, x_3) = \max_{x_4} f_1 + f_2 + f_4$
0	0	$14/x_4 = 1$
0	1	$14/x_4 = 1$
1	0	$17/x_4 = 1$
1	1	$17/x_4 = 1$
2	0	$15/x_4 = 1$

Bucket Elimination Example: Table Functions

Process bucket

$B(x_3) = \{f_3(x_2, x_3), f_5(x_1, x_2, x_3), f_6(x_1, x_3), g_{x_4}(x_2, x_3)\}$

x_2	x_3	f_3
0	0	3
0	1	3
1	0	0
1	1	8
2	0	2
2	1	1

x_1	x_2	x_3	f_5
0	0	0	3
0	0	1	6
0	1	0	6
0	1	1	5
0	2	0	4
0	2	1	3
1	0	0	2
1	0	1	1
1	1	0	3
1	1	1	5
1	2	0	4
1	2	1	2

x_1	x_3	f_6
0	0	5
0	1	4
1	0	5
1	1	6
2	0	4
2	1	2

Bucket Elimination Example: Table Functions

Process bucket

$B(x_3) = \{f_3(x_2, x_3), f_5(x_1, x_2, x_3), f_6(x_1, x_3), g_{x_4}(x_2, x_3)\}$

x_2	x_3	g_{x_4}
0	0	$14/x_4 = 1$
0	1	$14/x_4 = 1$
1	0	$17/x_4 = 1$
1	1	$17/x_4 = 1$
2	0	$15/x_4 = 1$
2	1	$15/x_4 = 1$

x_1	x_2	$g_{x_3} = \max_{x_3} f_3 + f_5 + f_6 + g_{x_4}$
0	0	$27/x_3 = 1$
1	0	$24/x_3 = 0$
2	0	$23/x_3 = 0$
0	1	$34/x_3 = 1$
1	1	$36/x_3 = 1$
2	1	$29/x_3 = 1$
0	2	$26/x_3 = 0$
1	2	$25/x_3 = 0$
2	2	$23/x_3 = 0$

Bucket Elimination Example: Table Functions

Process bucket $B(x_2) = \{g_{x_3}(x_1, x_2)\}$

x_1	x_2	$g_{x_3}(x_1, x_2)$
0	0	$27/x_3 = 1$
1	0	$24/x_3 = 0$
2	0	$23/x_3 = 0$
0	1	$34/x_3 = 1$
1	1	$36/x_3 = 1$
2	1	$29/x_3 = 1$
0	2	$26/x_3 = 0$
1	2	$25/x_3 = 0$
2	2	$23/x_3 = 0$

x_1	$g_{x_2}(x_1)$
0	$34/x_2 = 1$
1	$36/x_2 = 1$
2	$29/x_2 = 1$

Bucket Elimination Example: Table Functions

Process bucket $B(x_1) = \{g_{x_2}(x_1)\}$

x_1	$g_{x_2}(x_1)$
0	$34/x_2 = 1$
1	$36/x_2 = 1$
2	$29/x_2 = 1$

Maximum objective value

$g_{x_1}() = 36/x_1 = 1$

Bucket Elimination Example: Table Functions

Moving forward to find the optimal solution

Bucket Elimination Example: Table Functions

Moving forward to find the optimal solution

- $g_{x_1}() = 36 = \max_{x_1} g_{x_2}(x_1)$ was realized via $x_1 = 1$.

Bucket Elimination Example: Table Functions

Moving forward to find the optimal solution

- $g_{x_1}() = 36 = \max_{x_1} g_{x_2}(x_1)$ was realized via $x_1 = 1$.
- $g_{x_2}(1) = 36 = \max_{x_2} g_{x_3}(1, x_2)$ was realized via $x_2 = 1$.

Bucket Elimination Example: Table Functions

Moving forward to find the optimal solution

- $g_{x_1}() = 36 = \max_{x_1} g_{x_2}(x_1)$ was realized via $x_1 = 1$.
- $g_{x_2}(1) = 36 = \max_{x_2} g_{x_3}(1, x_2)$ was realized via $x_2 = 1$.
- $g_{x_3}(1, 1) = 36 =$

$$\max_{x_3} f_3(1, x_3) + f_5(1, 1, x_3) + f_6(1, x_3) + g_{x_4}(1, x_3)$$

  was realized via $x_3 = 1$.

Bucket Elimination Example: Table Functions

Moving forward to find the optimal solution

- $g_{x_1}() = 36 = \max_{x_1} g_{x_2}(x_1)$ was realized via $x_1 = 1$.
- $g_{x_2}(1) = 36 = \max_{x_2} g_{x_3}(1, x_2)$ was realized via $x_2 = 1$.
- $g_{x_3}(1, 1) = 36 =$

  $$\max_{x_3} f_3(1, x_3) + f_5(1, 1, x_3) + f_6(1, x_3) + g_{x_4}(1, x_3)$$

  was realized via $x_3 = 1$.
- $g_{x_4}(1, 1) = 17 =$

  $$\max_{x_4} f_1(x_4) + f_2(1, 1, x_4) + f_4(1, x_4)$$

  was realized via $x_4 = 1$.

Bucket Elimination Example: Table Functions

Moving forward to find the optimal solution

- $g_{x_1}() = 36 = \max_{x_1} g_{x_2}(x_1)$ was realized via $x_1 = 1$.
- $g_{x_2}(1) = 36 = \max_{x_2} g_{x_3}(1, x_2)$ was realized via $x_2 = 1$.
- $g_{x_3}(1, 1) = 36 =$

  $$\max_{x_3} f_3(1, x_3) + f_5(1, 1, x_3) + f_6(1, x_3) + g_{x_4}(1, x_3)$$

  was realized via $x_3 = 1$.
- $g_{x_4}(1, 1) = 17 =$

  $$\max_{x_4} f_1(x_4) + f_2(1, 1, x_4) + f_4(1, x_4)$$

  was realized via $x_4 = 1$.
- **Optimal solution:** $(1, 1, 1, 1)$

Using Bucket Elimination as Part of Branch and Bound

Computing Upper Bound with Buckets

Assume variable ordering $x_1, \ldots, x_n$, and objective function $F = \sum_i f_i$. Let a be an assignment over $x_1, \ldots, x_{n-k-1}$. Then $F_{\max}(a)$ may be approximated by summing over all bucket functions of BS_k, the bucket state that occurs after the k latest buckets are processed. This gives the formula

$$F_{\max}(a) = \sum_{i=1}^{n-k-1} \sum_{f \in B(x_i)} f(a),$$

where $B(x_i)$ is the associated with bucket state BS_k.

Using Bucket Elimination as Part of Branch and Bound

Rationale for Computing Upper Bound with Buckets

Since the latest k variables have yet to be instantiated, we may apply max operators to them (i.e. process their buckets) to create functions $g_{x_n}, \ldots, g_{x_{n-k}}$. Each of these functions is a function over a subset of $x_1, \ldots, x_{n-k-1}$, and hence can be evaluated by using assignment a. Moreover, each of these functions outputs the maximum attainable sum (of functions in the associated bucket). Finally, since any function f_i is either in one of the buckets $1, \ldots, n-k-1$ (and hence can be evaluated using a), or is in one of the k later buckets, it follows that f_i's contribution to $f_{\max}$ will be correctly recorded.

Using Bucket Elimination as Part of Branch and Bound

Example of Computing Upper Bound U with Buckets

Assume variable ordering and table functions from previous example. Let $a = (1, 0)$ be an assignment over $\{x_1, x_2\}$. Hence, $k = 2$ with BS_2 consisting of $B(x_1) = \emptyset$ and $B(x_2) = \{g_{x_3}(x_1, x_2)\}$. Then

$$F_{\max}(a) = g_{x_3}(1, 0) = 24.$$

Using Bucket Elimination as Part of Branch and Bound

Example of Computing Upper Bound U with Buckets

Assume variable ordering and table functions from previous example. Let $a = (1, 0)$ be an assignment over $\{x_1, x_2\}$. Hence, $k = 2$ with BS_2 consisting of $B(x_1) = \emptyset$ and $B(x_2) = \{g_{x_3}(x_1, x_2)\}$. Then

$$F_{\max}(a) = g_{x_3}(1, 0) = 24.$$

Why is the upper bound only an approximation?

Given assignment a over $x_1, \ldots, x_{n-k-1}$, one can **move forward** to compute an assignment $a \cup b$ over $x_1, \ldots, x_n$ that realizes $F_{\max}$, where b is an assignment over $x_{n-k}, \ldots, x_n$. However, it is possible that $a \cup b$ does not satisfy all the hard constraints. Hence, in the presence of hard constraints, there is no guarantee that $F_{\max}$ can be realized.

Complexity of Bucket Elimination

Bucket processing grows exponentially with number of variables

Let d be a bound on the size of each variable domain. Let n be the number of variables y for which $y \neq x$ and $y \in \text{var}(f)$, for some $f \in B(x)$. Then g_x has domain size d^n.

Complexity of Bucket Elimination

Managing Complexity with Upper Approximations

Complexity of Bucket Elimination

Managing Complexity with Upper Approximations

- h is called an **upper approximation** of g iff $g(z) \leq h(z)$, for all domain values z of g.

Complexity of Bucket Elimination

Managing Complexity with Upper Approximations

- h is called an **upper approximation** of g iff $g(z) \leq h(z)$, for all domain values z of g.
- **Upper approximation via** max **operator distribution.** If $g(u, v, w, x, y, z, t) = \max_t f_1(u, v, w, t) + f_2(x, y, z, t)$, then $g(z) \leq h_1(u, v, w) + h_2(x, y, z) = \max_t f_1(u, v, w, t) + \max_t f_2(x, y, z, t)$. Therefore h_1 and h_2 have smaller domains, and require less computation than g.

Complexity of Bucket Elimination

Managing Complexity with Upper Approximations

- h is called an **upper approximation** of g iff $g(z) \leq h(z)$, for all domain values z of g.
- **Upper approximation via** max **operator distribution.** If $g(u, v, w, x, y, z, t) = \max_t f_1(u, v, w, t) + f_2(x, y, z, t)$, then $g(z) \leq h_1(u, v, w) + h_2(x, y, z) = \max_t f_1(u, v, w, t) + \max_t f_2(x, y, z, t)$. Therefore h_1 and h_2 have smaller domains, and require less computation than g.
- The use of max operator distribution for bucket processing is called **mini-bucket elimination**. Here, processed bucket $B(x)$ may yield several functions with lower complexity than g_x. However, these functions are upper-approximations of g_x. So exact solutions are no longer guaranteed.