I214 System Optimization
Chapter 6: Combinatorial Optimization (Part C)

Brian Kurkoski

Japan Advanced Institute of Science and Technology

2023 January

Outline

6.3 Dynamic Programming

Dynamic programming solves an optimization problem by dividing the main problem into smaller subproblems.

If it is possible to find an optimal solution to the subproblems, then this can be used to find the optimal solution of the main problem.

A problem exhibits optimal substructure if the optimal solution contains optimal solutions to subproblems.

To find a dynamic programming method for solving a specific problem:

1. Characterize the structure of the optimal solution.
2. Recursively define the value of the optimal solution.
3. Compute the optimal value.
4. Construct the optimal solution.

Examples

Examples of problems solved with dynamic programming:

- Rod cutting — Subproblem is optimally cutting rods of a shorter length
- Matrix multiplication — Subproblem is optimal parenthesizations of matrix product for i to j
- Knapspack problem

6.3.4 Knapsack Problem

Recall the Knapsack problem:

Problem

Given a set of n items, each with given weight a_i and utility $c_i \geq 0$, select the items to maximize the total utility while having total weight not greater than W.

In this case, there is only one of each item, that is $x_i \in \{0, 1\}$, where x_i is the number of item i in the knapsack.

Apply Dynamic Programming Principles to Knapsack Problem

Dynamic programming principles.

To find a dynamic programming method for solving a specific problem:

1. Characterize the structure of the optimal solution.
2. Recursively define the value of the optimal solution.
3. Compute the optimal value.
4. Construct the optimal solution.

Example

Use dynamic programming to solve the $n = 4$ knapsack problem with

$$(c_1, c_2, c_3, c_4) = (7, 8, 1, 2)$$
$$(a_1, a_2, a_3, a_4) = (4, 5, 1, 3)$$

and $W = 6$.

Computation of $p(i, w)$ for Example

$p(i,w)$ / $x(i,w)$	$w=0$	$w=1$	$w=2$	$w=3$	$w=4$	$w=5$	$w=6$
$i=0$	0 / 0	0 / 0	0 / 0	0 / 0	0 / 0	0 / 0	0 / 0
$i=1$	0 / 0	0 / 0	0 / 0	0 / 0	7 / 1	7 / 1	7 / 1
$i=2$	0 / 0	0 / 0	0 / 0	0 / 0	7 / 0	8 / 1	8 / 1
$i=3$	0 / 0	1 / 1	1 / 1	1 / 1	7 / 0	8 / 0	9 / 1
$i=4$	0 / 0	1 / 0	1 / 0	2 / 1	7 / 0	8 / 0	9 / 0

Construct Optimal Solution for Example

$p(i,w)$ / $x(i,w)$	$w=0$	$w=1$	$w=2$	$w=3$	$w=4$	$w=5$	$w=6$
$i=0$	0 / 0	0 / 0	0 / 0	0 / 0	0 / 0	0 / 0	0 / 0
$i=1$	**0 / 0**	0 / 0	0 / 0	0 / 0	7 / 1	7 / 1	7 / 1
$i=2$	0 / 0	0 / 0	0 / 0	0 / 0	7 / 0	**8 / 1**	8 / 1
$i=3$	0 / 0	1 / 1	1 / 1	1 / 1	7 / 0	8 / 0	**9 / 1**
$i=4$	0 / 0	1 / 0	1 / 0	2 / 1	7 / 0	8 / 0	**9 / 0**

Class Info

- Homework 7 (KKT conditions). Deadline: February 3 at 18:00
- Homework 8 (dynamic programming). Do not submit. Solutions will be published before final exam.
- Final lecture: Monday, February 6. Review course.
- Final exam: Wednesday, February 8 at 13:30–15:10.

Final Exam

The midterm is Wednesday, February 8 at 13:30–15:10. The exam is closed book. You may use:

- One page of notes, A4-sized paper, double-sided OK.
- Blank scratch paper

You may not use anything else: No printed materials, including books, lecture notes, and slides. No notes (except as above). No internet-connected devices. No calculators.

Exam Content

- Covers Chapters 5 and 6 only
- Differs from midterm: not all numerical problems, some analytical problems
- Problems similar to Homework 6–8. Solutions to Homework 6–8 are provided.
- Additional exercises to practice for exam:
  - Example 5.9 (try to solve without looking at solution)
  - Section 6.1.3, Exercise [1]
  - Section 6.3.4, Exercise [1]
  - Section 6.3.4, Exercise [2]