Supermarket Sweep

Adam Profili, Brynn Schneberger

4/25/2021

Project Overview

A contestant in a gameshow has a default of 90 seconds to make their way around a grocery store, collect up to a default of 15 items in their cart, and return to the start.

Their score is the total cost of the items in their cart. Items are spread around the pictured aisles, and each have specific prices and locations.

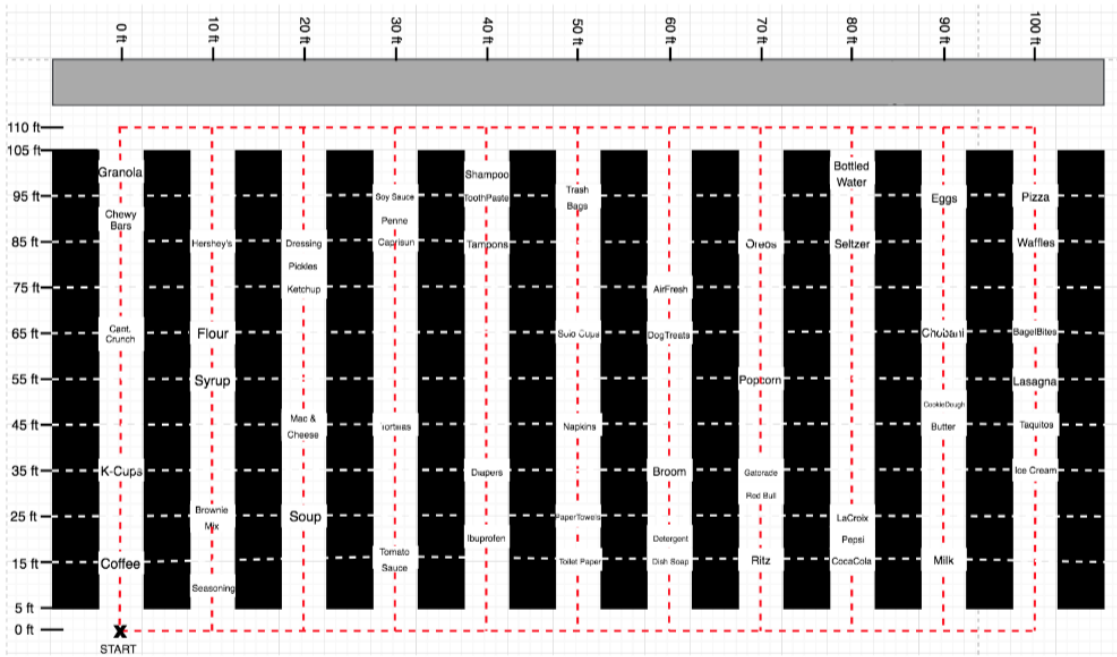


Figure 1: Grocery Store Layout

(a) asks:

For each pair of items $i, j \in [\![1, 56]\!]$, compute the shortest time d_{ij} to go from item i to item j, and store these values. You may not use any optimization software to answer this question. Compute also the shortest time to travel between the start/end location and any item.

(a) solution:

If two items are on the same x-aisle, the shortest distance from one to the other is the absolute value of the difference between their y-coordinates. We then divide the distance by the contestant's speed, 10 feet/second to convert this calculation to time.

If two items are on different x-aisle, the shortest distance from one to the other is either by moving up to the end of the aisle, across to the other item's aisle, and then down to the item, or by moving down to the start of the aisle, across to the other item's aisle, and then up to the item. An algorithm should choose the shortest of these two options, which could be done by taking the minimum of the two or by finding if the two item's y-values on average are closer to the end (110 ft) or the beginning (0 ft). We then divide the distance by the contestant's speed, 10 feet/second to convert this distance to time.

(a) solution:

Refer to the code

(b) asks:

Formulate the Supermarket Sweep problem as a Mixed-Integer Program.

(b) Data Placeholders:

n represents the number of nodes, with node 1 being the start node, nodes 2,3,. . . ,n being item nodes, and n+1 being the end node which shares the attributes of the start node.

T represents the maximum time the contestant is given to shop.

C represents the maximum amount of items the contestant can put in their cart.

v_i represents the value of node i. $\forall i = 1, 2, ..., n$

d_{ij} represents the minimum time it takes to move from node i to node j. If node j represents an item and not a start/end point, it will include the 2 seconds to add that item to the cart.
$\forall i = 1, 2, ..., n \; \forall j = 2, 3, ..., n + 1$

(b) Decision Variables:

$x_{ij} = 1$ if node j follows node i in the chosen path, 0 otherwise.
$\forall i = 1, 2, ..., n \; \forall j = 2, 3, ..., n + 1$

$y_j = 1$ if node j follows node i in the chosen path, 0 otherwise.
$\forall j = 1, 2, ..., n + 1$

$t_{ij} = y_j$ if node j follows node i in the chosen path, 0 otherwise.
$\forall i = 1, 2, ..., n \; \forall j = 2, 3, ..., n + 1$

(b) Objective:

$$\max_{x,y,t} \quad \text{score} = \sum_{i=1}^{n} v_i \sum_{j=2}^{n+1} x_{ij}$$

Maximize the value of all items picked up.

(b) Constraint (1)

$$y_1 = 0$$

(1) The path begins at node 1 with a time of 0.

(b) Constraint (2)

$$\sum_{j=2}^{n+1} x_{1,j} = 1$$

(2) Node 1 has exactly one destination node directly after it in the path.

(b) Constraint (3)

$$\sum_{j=2}^{n+1} x_{ij} \leq 1 \quad \forall i \in [\![2, n]\!]$$

(3) Nodes 2 through n may have at most one destination node directly after it in the path.

(b) Constraint (4)

$$\sum_{j=2}^{n+1} x_{ij} \leq 1 \quad \forall i \in [\![2, n]\!]$$

(4) Nodes 2 through n may have at most one origin node directly before it in the path.

(b) Constraint (5)

$$\sum_{i=1}^{n} x_{ij} \leq 1 \quad \forall j \in [\![2, n]\!]$$

(4) Nodes 2 through n may have at most one origin node directly before it in the path.

(b) Constraint (5)

$$\sum_{i=1}^{n} x_{i,n+1} = 1$$

(5) Node n+1 has exactly one origin node directly before it in the path.

(b) Constraint (6)

$$t_{ij} \leq T x_{ij} \quad \forall i \in [\![1, n]\!] \qquad \forall j \in [\![2, n+1]\!]$$

(6) If the direct path from node i to node j exists, than $t_i j$ is upper bounded at T. Otherwise, it is constrained to equal 0.

(b) Constraint (7)

$$y_j = \sum_{i=1}^{n} t_{ij} \quad \forall j \in [\![2, n+1]\!]$$

(7) Each running time y_j is set as the sum over all i of t_ij for all destination nodes j, of which at most one is nonzero.

(b) Constraint (8)

$$\sum_{k=2}^{n+1} t_{jk} = y_j + \sum_{k=2}^{n+1} d_{jk} x_{jk} \quad \forall j \in [\![1, n]\!]$$

(8) We define the sum over destination nodes k of $t_j k$ as the sum of the running total at node j plus the additional time added by the chosen destination.

(b) Constraint (9)

$$x_{ii} = 0 \quad \forall i \in [\![1, n]\!]$$

(9) No node may follow itself in the path.

(b) Constraint (10)

$$\sum_{i=1}^{n} x_{ij} = \sum_{k=2}^{n+1} x_{jk} \quad \forall j \in [\![2, n]\!]$$

(10) Nodes that are not entered may not be exited, and nodes that are entered must be exited.

(b) Constraint (11)

$$\sum_{i=1}^{n} \sum_{j=2}^{n} x_{ij} \leq C$$

(11) The amount of item nodes in the path must be at most C.

(b) Bound Constraints

$$x_{ij} \in \{0,1\} \quad \forall i \in [\![1,n]\!] \qquad \forall j \in [\![2,n+1]\!]$$

$$t_{ij} \geq 0 \qquad \forall i \in [\![1,n]\!] \qquad \forall j \in [\![2,n+1]\!]$$

(c) asks:

Code and solve your optimization model. What is the optimal path? Which items are picked? What is the total value of these items?

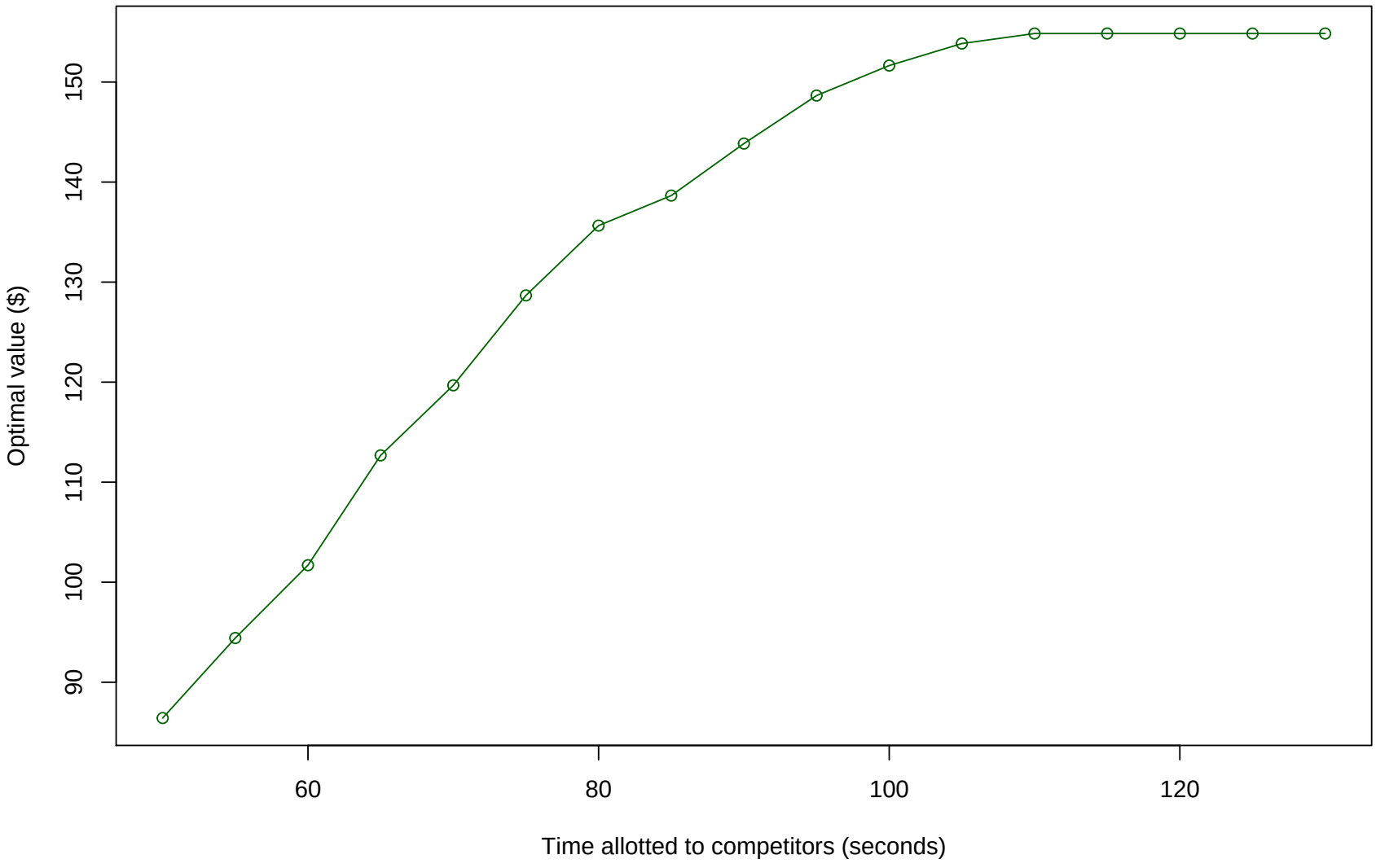
(c) solution:

Refer to code output

(d) asks:

How does the optimal value of the problem vary with respect to the total amount of time allowed (initially set to 90 sec)? Answer this question by plotting and analyzing the optimal value of the problem with different values of the total amount of time that is allowed.

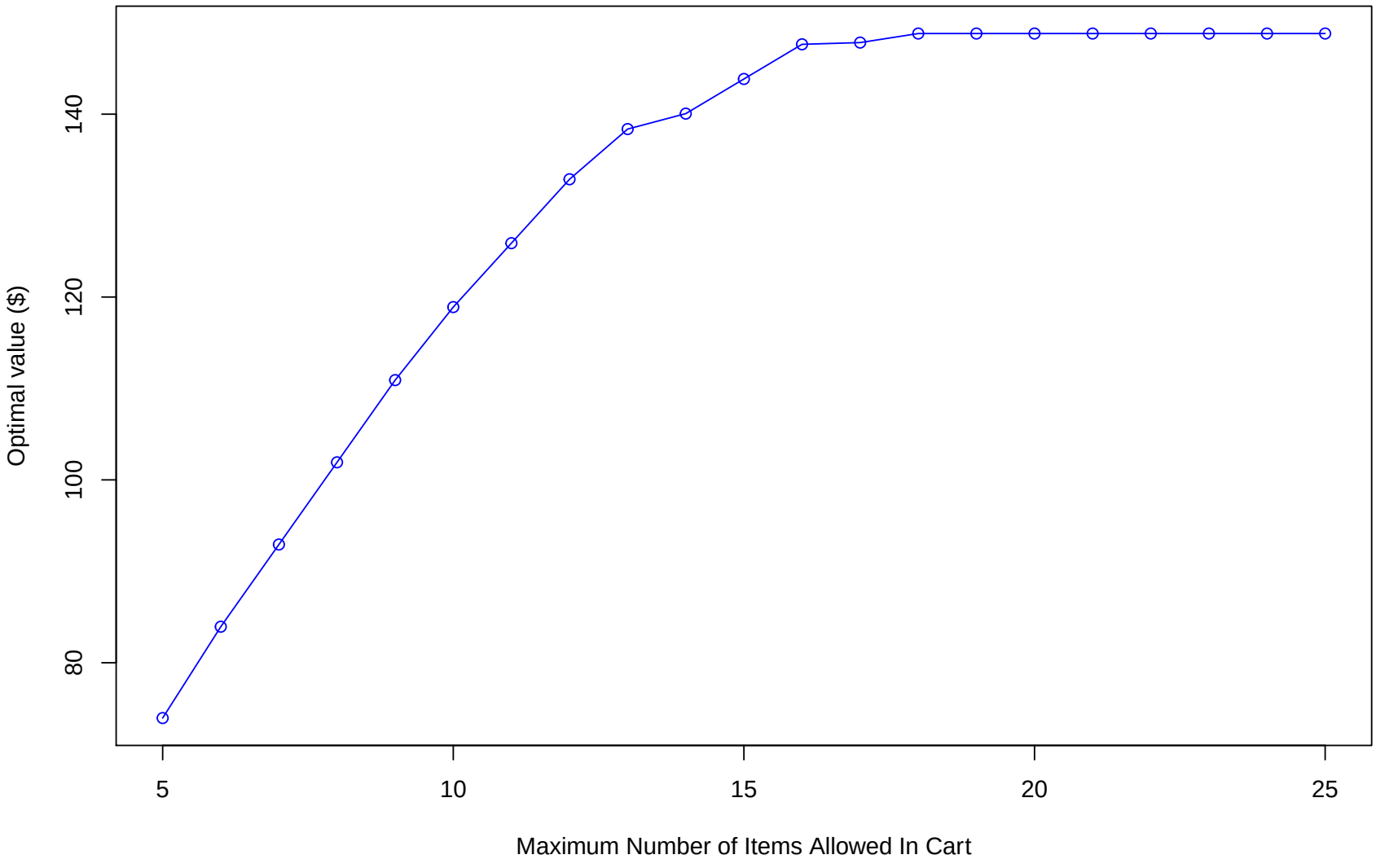
(d) solution:


Optimal value ($)
90
100
110
120
130
140
150
60
80
100
120
Time allotted to competitors (seconds)

(e) asks:

How does the optimal value of the problem vary with respect to the capacity of the cart (initially set to 15)? Answer this question by plotting and analyzing the optimal value of the problem with different values of the cart capacity.

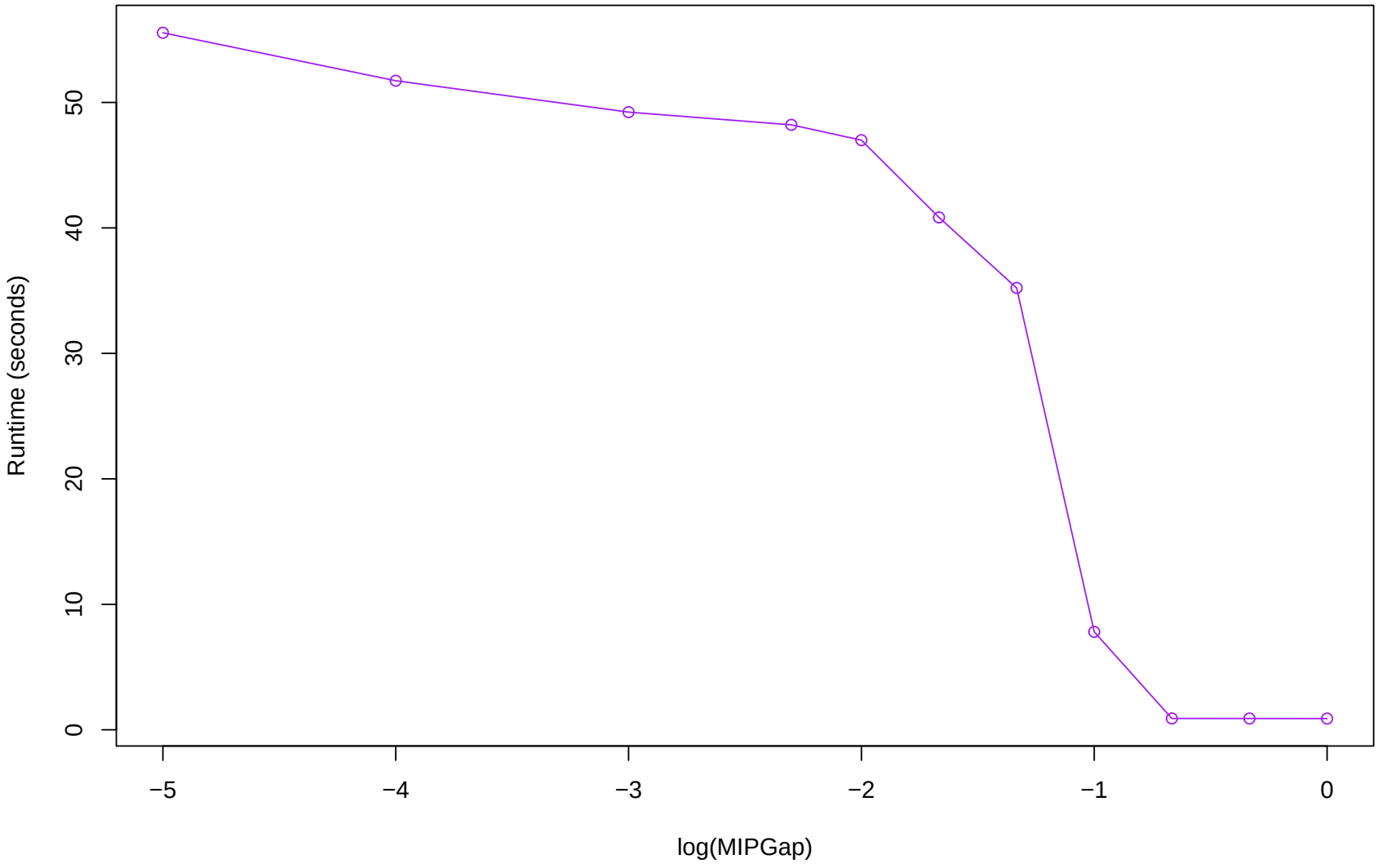
(e) solution:


Optimal value ($)
80
100
120
140
5
10
15
20
25
Maximum Number of Items Allowed In Cart

(f) asks:

Consider the initial parameters of the problem (cart of size 15, time allowed of 90 seconds). Gurobi has a parameter called MIPGap. Its default value is 0.0001. This means that Gurobi will output a solution, as soon as its objective value is within 0.01% of the optimal value of the problem. Changing this parameter will influence the time it takes Gurobi to output a solution. What happens if you change this Gurobi parameter for the Supermarket Sweep problem? Answer this question by plotting and analyzing the time Gurobi takes to output a solution with respect to the optimality gap that is allowed.

(f) solution:



Conclusion

Thank you for listening to our presentation.