



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ
ΜΗΧΑΝΙΚΩΝ ΚΑΙ
ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ

OPTIMAL DIET PROBLEM

ΓΡΑΜΜΙΚΗ ΚΑΙ ΣΥΝΔΥΑΣΤΙΚΗ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ

ΝΑΤΑΛΙΑ ΡΟΥΣΚΑ - AM 1092581

ΥΠΕΥΘΥΝΗ ΚΑΘΗΓΗΤΡΙΑ

ΣΟΦΙΑ ΔΑΣΚΑΛΑΚΗ

ΡΙΟ

22 Σεπτεμβρίου 2025

A. Εισαγωγή

Στη σύγχρονη εποχή, πέρα από τις διατροφικές απαιτήσεις, σημαντικό ρόλο παίζουν και οι οικονομικοί περιορισμοί, συνεπώς ο σχεδιασμός μίας δίαιτας που θα τα καλύπτει και τα δύο οδηγεί σε προβλήματα βελτιστοποίησης.

Η παρούσα εργασία βασίζεται στο πρόβλημα διαίτας όπως διατυπώθηκε από τον Stigler, το 1945. Συγκεκριμένα, για έναν δραστήριο άντρα περίπου 70 κιλών, να προσδιοριστούν οι ποσότητες από 77 φαγητά ώστε να καλύπτονται τα ελάχιστα επίπεδα ημερήσιας πρόσληψης 9 θρεπτικών συστατικών (Recommended Dietary Allowances – RDAs) με το ελάχιστο δυνατό οικονομικό κόστος. Τα συστατικά με τα οποία ασχολήθηκε ο Stigler είναι θερμίδες, πρωτεΐνη, ασβέστιο, σίδηρος, βιταμίνη A, θειαμίνη, ριβοφλαβίνη, νιασίνη και ασκορβικό οξύ.

Ο Stigler χρησιμοποίησε προσεγγιστικές μεθόδους δοκιμής και λάθους για να λύσει το πρόβλημα και κατέληξε σε μία δίαιτα με ημερήσιο κόστος 0.109\$ εν έτει 1940. Αργότερα, το πρόβλημα λύθηκε με τη μέθοδο Simplex και βρέθηκε η ακριβής βέλτιστη λύση πολύ κοντά σε αυτή που είχε διατυπωθεί νωρίτερα.

B. Περίληψη

Σκοπός της εργασίας είναι να λυθεί ένα παρόμοιο πρόβλημα με γραμμικό προγραμματισμό που προσδιορίζει μία δίαιτα ανάμεσα σε 64 φαγητά και καλύπτει τις ημερίσιες ανάγκες σε 8 θρεπτικά συστατικά (συμπεριλαμβανομένων των θερμίδων) με το ελάχιστο δυνατό κόστος.

Παρουσιάζονται δύο σενάρια επίλυσης. Στο πρώτο, βρίσκεται η βέλτιστη λύση που αφορά την πρώτη μέρα και για κάθε επόμενη λύνουμε το πρόβλημα ξανά εισάγοντας περιορισμούς αποκλεισμού των τροφίμων που προτάθηκαν για την προηγούμενη ημέρα κ.ο.κ.

Στο δεύτερο σενάριο λύνουμε ένα πρόβλημα γ.π. για όλες τις ημέρες ταυτόχρονα, στο οποίο εισάγονται εξ αρχής περιορισμοί εναλλαγής των τροφίμων μεταξύ των ημερών.

Τα διατροφικά δεδομένα, δηλαδή ποσότητα θρεπτικών συστατικών ανά τροφή και ημερίσιες απαιτήσεις σε θρεπτικά συστατικά προέρχονται από πίνακες USDA (United States Department of Agriculture), όπως έχουν οργανωθεί στο case study

[1]. Παρακάτω φαίνονται ενδεικτικά μερικά δεδομένα από τους πίνακες, όπως έχουν περαστεί σε .xlsx αρχεία για να χρησιμοποιηθούν στο μοντέλο.

Foods	Price/ Serving	Serving Size	Calories	Cholesterol mg	Total_Fat g	Sodium mg	Carbohydrates g	Dietary_Fiber g	Protein g	Vit_A IU	Vit_C IU	Calcium mg	Iron mg	Price/ Serving g New
Frozen Broccoli	\$0,16	10 Oz Pkg	73,8	0	0,8	68,2	13,6	8,5	8	5867	160,2	159	2,3	0,64
Carrots, Raw	\$0,07	1/2 Cup Shredded	23,7	0	0,1	19,2	5,6	1,6	0,6	15471	5,1	14,9	0,3	0,28
Celery, Raw	\$0,04	1 Stalk	6,4	0	0,1	34,8	1,5	0,7	0,3	53,6	2,8	16	0,2	0,16

Figure 1: Amount of nutrition/price of each food

	Calories	Carbohy drates g	Dietary_ Fiber g	Protein g	Vit_A IU	Vit_C IU	Calcium mg	Iron mg
Minimum daily intake	2000	0	25	50	5000	50	800	10
Maximum daily intake	2250	300	100	100	50000	20000	1600	30

Figure 2: Daily required nutrition intakes

Το πρόβλημα, όπως μοντελοποιείται και στα δύο σενάρια λύνεται σε Python με τη βιβλιοθήκη pulp και τον CBC solver.

C. Σενάριο 1- Μοντελοποίηση προβλήματος

Παράμετροι
$Foods: i \in \{1, \dots, M\}$
$Nutrients: n \in \{1, \dots, N\}$
c_i : κόστος μερίδας φαγητού i
a_{ni} : ποσότητα συστατικού n σε μία μερίδα φαγητού i
N_n^{min} : ελάχιστη ημερίσια ποσότητα συστατικού n
N_n^{max} : μέγιστη ημερίσια ποσότητα συστατικού n
S : μέγιστος αριθμός μερίδων κάθε τροφής ημερισίως

Μεταβλητές απόφασης
x_i : αριθμός μερίδων φαγητού i
y_i : δυαδική μεταβλητή για την επιλογή της τροφής i
$y_i = 1$, αν $x_i > 0$
$y_i = 0$, αν $x_i = 0$

Αντικειμενική συνάρτηση $\min z = \sum_{i=1}^M c_i x_i$

Περιορισμοί
Για κάθε συστατικό n $N_n^{min} < \sum_{i=1}^M a_{ni} x_i < N_n^{max}$
$x_i \geq 0 \forall i$ και ακέραιοι

$x_i \leq S y_i \forall i, \text{αν } y_i = 0 \text{ τότε } x_i = 0 \text{ (μη επιλεγμένη τροφή)}$ $x_i \geq y_i \forall i, \text{αν } y_i = 1 \text{ τότε } 1 \leq x_i \leq S \text{ (επιλεγμένη τροφή με τουλάχιστον μία μερίδα)}$

Περιορισμός που εισάγεται για διαφοροποίηση στις επόμενες λύσεις Για κάθε προηγούμενο σετ τροφών $C \subseteq \{1, \dots, M\} : \sum_{i \in C} y_i \leq C - k$, όπου k ο αριθμός των τροφών που πρέπει να αλλάξουν σε σχέση με την κάθε προηγούμενη μέρα 2 ^η μέρα (+1περιορισμός) ... 7 ^η μέρα(+6 περιορισμοί)
--

D. Σενάριο 1 - Υλοποίηση

Στο συγκεκριμένο πρόβλημα έχουμε M=64 τρόφιμα και N=8 συστατικά (θερμίδες, πρωτεΐνη, ασβέστιο, σίδηρος, βιταμίνη A, βιταμίνη C, υδατάνθρακες και φυτικές ίνες). Τις τιμές των παραμέτρων $c_i, a_{ni}, N_n^{min}, N_n^{max}$ τις παίρνουμε από τους πίνακες και λύνουμε για κάθε μια ημέρα ένα ξεχωριστό πρόβλημα ακέραιου γ.π. Σε μία πρώτη εκδοχή του προβλήματος ορίζουμε S=2, k=8.

Για την πρώτη ημέρα διατυπώνεται το πρόβλημα ακέραιου γ.π όπως φαίνεται παρακάτω στον κώδικα και έχουμε βέλτιστη λύση.

```

prob = pulp.LpProblem(f"diet_day{run}", pulp.LpMinimize)

x = pulp.LpVariable.dicts('x', range(M), lowBound=0, cat=pulp.LpInteger) ##number of servings in each food
y = pulp.LpVariable.dicts('y', range(M), cat=pulp.LpBinary) ##1 if food is chosen else 0

prob += pulp.lpSum([costs[i] * x[i] for i in range(M)])

# Nutrition Constraints
for i in range(N):
    prob += pulp.lpSum([A[i][j] * x[j] for j in range(M)]) >= Nmin[i], f'min_nutrient_{i}'
    prob += pulp.lpSum([A[i][j] * x[j] for j in range(M)]) <= Nmax[i], f'max_nutrient_{i}'

for i in range(M):
    prob += x[i] <= S * y[i], f'link_upper_{i}' ##if y[i]=0 then x[i]=0 if y[i]=1 x[i]<=S
    prob += x[i] >= y[i], f'link_lower_{i}' ## if y[i]=1 then x[i]>=1
  
```

Figure 3: Μοντελοποίηση του προβλήματος σε κώδικα

```

---Day #1 ---
Total Cost: €6.80
Active foods (indices): [0, 1, 6, 12, 15, 17, 18, 21, 24, 27, 48, 52]
Frozen Broccoli: 2.00 servings
Carrots,Raw: 2.00 servings
Banana: 1.00 servings
Oranges: 1.00 servings
Wheat Bread: 2.00 servings
White Bread: 2.00 servings
Chocolate Chip Cookies: 2.00 servings
3.3% Fat,Whole Milk: 1.00 servings
Poached Eggs: 2.00 servings
Peanut Butter: 2.00 servings
Popcorn,Air-Popped: 2.00 servings
Potatoes, Baked: 2.00 servings
  
```

Figure 4: Βέλτιστη λύση για την 1η ημέρα

Για κάθε επόμενη ημέρα εισάγουμε έναν περιορισμό διαφοροποίησης για ποικιλία τροφών ανάμεσα στις ημέρες. Συγκεκριμένα, στη λίστα `used_combinations` αποθηκεύουμε τους συνδυασμούς τροφών που προτάθηκαν τις προηγούμενες μέρες και εισάγουμε τόσους περιορισμούς όσοι οι προηγούμενες μέρες ώστε η τρέχουσα μέρα να διαφέρει τουλάχιστον k τροφές από την κάθε προηγούμενη (το πολύ $|C| - k$ τροφές να ξαναεμφανιστούν $\forall C$).

```
for combo in used_combinations:
    prob += pulp.lpSum([y[i] for i in combo]) <= len(combo) - k #at least k foods changed
```

Figure 5: Περιορισμός διαφοροποίησης

Παρακάτω φαίνεται η βέλτιστη λύση για την κάθε ημέρα. Για παράδειγμα, στις δύο πρώτες ημέρες παραμένουν ίδιες οι τροφές $[0, 6, 48, 52]$. Ανάμεσα στην 2^η- 3^η μέρα παραμένουν ίδιες οι $[0, 52]$ και ανάμεσα στην 1^η – 3^η οι $[0, 12, 15, 52]$ κ.ο.κ

```
---Day #2 ---
Total Cost: €8.32
Active foods (indices): [0, 6, 9, 22, 26, 28, 46, 48, 52, 55]
Frozen Broccoli: 2.00 servings
Butter,Regular: 1.00 servings
Skim Milk: 1.00 servings
Scrambled Eggs: 1.00 servings
White Rice: 2.00 servings
Peanut Butter: 2.00 servings
Popcorn,Air-Popped: 2.00 servings
Tortilla Chip: 1.00 servings
Potatoes, Baked: 2.00 servings
Spaghetti W/ Sauce: 1.00 servings
```

```
---Day #3 ---
Total Cost: €12.32
Active foods (indices): [0, 2, 3, 4, 11, 12, 15, 19, 25, 49, 52, 54]
Frozen Broccoli: 2.00 servings
Apple,Raw,W/Skin: 2.00 servings
Banana: 2.00 servings
Oranges: 2.00 servings
Oatmeal Cookies: 2.00 servings
Celery, Raw: 1.00 servings
2% Lowfat Milk: 1.00 servings
Frozen Corn: 1.00 servings
Lettuce,Iceberg,Raw: 2.00 servings
Pork: 1.00 servings
Popcorn,Air-Popped: 2.00 servings
Pretzels: 1.00 servings
```

```
---Day #4 ---
Total Cost: €13.12
Active foods (indices): [0, 2, 9, 17, 18, 19, 21, 23, 32, 53, 54]
Frozen Broccoli: 2.00 servings
Wheat Bread: 2.00 servings
White Bread: 2.00 servings
Oatmeal Cookies: 2.00 servings
Celery, Raw: 2.00 servings
Chocolate Chip Cookies: 2.00 servings
Cheddar Cheese: 1.00 servings
Kielbasa,Prk: 1.00 servings
Potato Chips,Bbqflvr: 1.00 servings
Pretzels: 2.00 servings
Spaghetti W/ Sauce: 2.00 servings
```

```

---Day #5 ---
Total Cost: €13.20
Active foods (indices): [0, 11, 17, 24, 36, 47, 49, 52, 63]
  Frozen Broccoli: 2.00 servings
  Apple,Raw,W/Skin: 2.00 servings
  Wheat Bread: 2.00 servings
  3.3% Fat,Whole Milk: 1.00 servings
  Raisin Brn, Kellg'S: 1.00 servings
  Macaroni,Ckd: 2.00 servings
  Pork: 1.00 servings
  Popcorn,Air-Popped: 2.00 servings
  Beanbacn Soup,W/Watr: 1.00 servings

---Day #6 ---
Total Cost: €14.08
Active foods (indices): [0, 9, 12, 15, 21, 25, 34, 44]
  Frozen Broccoli: 2.00 servings
  Banana: 2.00 servings
  Oranges: 1.00 servings
  Chocolate Chip Cookies: 2.00 servings
  2% Lowfat Milk: 1.00 servings
  Cheerios: 1.00 servings
  Hotdog, Plain: 2.00 servings
  Spaghetti W/ Sauce: 2.00 servings

---Day #7 ---
Total Cost: €18.68
Active foods (indices): [1, 2, 3, 4, 6, 14, 18, 23, 48, 57, 63]
  Carrots,Raw: 2.00 servings
  Kiwifruit,Raw,Fresh: 1.00 servings
  White Bread: 2.00 servings
  Celery, Raw: 2.00 servings
  Cheddar Cheese: 2.00 servings
  Frozen Corn: 2.00 servings
  Lettuce,Iceberg,Raw: 2.00 servings
  Peanut Butter: 2.00 servings
  Splt Pea&Hamsoup: 2.00 servings
  Potatoes, Baked: 2.00 servings
  Beanbacn Soup,W/Watr: 2.00 servings

```

Figure 6: Βέλτιστη λύση/ημέρα

	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7
Κόστος	Κόστος: €6.80	Κόστος: €8.32	Κόστος: €12.32	Κόστος: €13.12	Κόστος: €13.20	Κόστος: €14.08	Κόστος: €18.68
Μερίδες	Frozen Broccoli (2) Carrots,Raw (2) Banana (1) Oranges (1) Wheat Bread (2) White Bread (2) Chocolate Chip Cookies (2) 3.3% Fat,Whole Milk (1) Poached Eggs (2) Peanut Butter (2) Popcorn,Air-Popped (2) Potatoes, Baked (2)	Frozen Broccoli (2) Butter,Regular (1) Skim Milk (1) Scrambled Eggs (1) White Rice (2) Peanut Butter (2) Popcorn,Air-Popped (2) Tortilla Chip (1) Potatoes, Baked (2) Spaghetti W/ Sauce (1)	Frozen Broccoli (2) Apple,Raw,W/Skin (2) Banana (2) Oranges (2) Oatmeal Cookies (2) Celery, Raw (1) 2% Lowfat Milk (1) Frozen Corn (1) Lettuce,Iceberg,Raw (2) Pork (1) Popcorn,Air-Popped (2) Pretzels (1)	Frozen Broccoli (2) Wheat Bread (2) White Bread (2) Oatmeal Cookies (2) Celery, Raw (2) Chocolate Chip Cookies (2) Cheddar Cheese (1) Kielbasa,Prk (1) Potato Chips,Bbqflvr (1) Pretzels (2) Spaghetti W/ Sauce (2)	Frozen Broccoli (2) Apple,Raw,W/Skin (2) Wheat Bread (2) 3.3% Fat,Whole Milk (1) Raisin Brn, Kellg'S (1) Macaroni,Ckd (2) Pork (1) Popcorn,Air-Popped (2) Beanbacn Soup,W/Watr (1)	Frozen Broccoli (2) Banana (2) Oranges (1) Chocolate Chip Cookies (2) 2% Lowfat Milk (1) Cheerios (1) Hotdog, Plain (2) Spaghetti W/ Sauce (2)	Carrots,Raw (2) Kiwifruit,Raw,Fresh (1) White Bread (2) Celery, Raw (2) Cheddar Cheese (2) Frozen Corn (2) Lettuce,Iceberg,Raw (2) Peanut Butter (2) Splt Pea&Hamsoup (2) Potatoes, Baked (2) Beanbacn Soup,W/Watr (2)

Figure 7: Total plan

Ε. Σενάριο 1- Σχολιασμός των αποτελεσμάτων και ανάλυση για τις παραμέτρους S, k

Μία αρχική παρατήρηση είναι η σταδιακή αύξηση του κόστους. Τη πρώτη μέρα επιλέγεται το πιο φτηνό και θρεπτικό πλάνο. Από τη δεύτερη μέρα και μετά το μοντέλο αναγκάζεται να βρει πιο ακριβές επιλογές λόγω του αποκλεισμού κάποιων τροφών για να καλύψει τα θρεπτικά όρια.

Επιπλέον, μετά από σύγκριση των τροφών που παραμένουν ίδιες ανάμεσα στις μέρες παρατηρώ ότι το Frozen Broccoli (τροφή 0) παραμένει σχεδόν σταθερό. Αυτό σημαίνει ότι είναι αναγκαία για να μείνει χαμηλό το κόστος και να καλυφθούν τα θρεπτικά όρια, έχει δηλαδή υψηλό λόγο αξίας/κόστους.

Με $S=2$: Αν το k αυξηθεί σε 9 παίρνουμε λύση για τις 3 πρώτες ημέρες, όμως για την 4^η το πρόβλημα γίνεται ανέφικτο. Αν όμως μειωθεί σε 7, το μοντέλο ξαναχρησιμοποιεί περισσότερα από τα φθηνά/θρεπτικά τρόφιμα και η ποικιλία μειώνεται δραματικά ανάμεσα στις ημέρες. Το κόστος όμως δεν αυξάνεται πολύ από μέρα σε μέρα και ενδεικτικά την τελευταία μέρα έχουμε βέλτιστη λύση με 12.48\$.

Με $k=8$: Αν το S μειωθεί σε 1, το κόστος αυξάνεται ήδη από την πρώτη μέρα σε 11.52\$ γιατί χρειάζονται πολλά διαφορετικά τρόφιμα να καλύψουν τα θρεπτικά όρια, περίπου 15. Αν το S αυξηθεί σε 3, το μοντέλο κολλάει σε πολύ λίγα φθηνά/θρεπτικά τρόφιμα και με $k=8$ δεν μπορεί να δώσει λύση πέρα από τη πρώτη μέρα

Τελικά, αφού οι παράμετροι k , S είναι αλληλένδετες μία ρεαλιστική ισορροπία ώστε να έχουμε βέλτιστη λύση και για τις 7 ημέρες, να υπάρχει διαφοροποίηση αλλά και λογικό κόστος είναι $S=2$, $k=8$.

Φ. Σενάριο 2 – Μοντελοποίηση προβλήματος

Παράμετροι
$Foods: i \in \{1, \dots, M\}$
$Nutrients: n \in \{1, \dots, N\}$
c_i : κόστος μερίδας φαγητού i
a_{ni} : ποσότητα συστατικού n σε μία μερίδα φαγητού i
N_n^{min} : ελάχιστη ημερίσια ποσότητα συστατικού n
N_n^{max} : μέγιστη ημερίσια ποσότητα συστατικού n
S : μέγιστος αριθμός μερίδων κάθε τροφής ημερισίως
$Days: d \in \{1, \dots, D\}$

Μεταβλητές απόφασης
$x_{i,d}$: αριθμός μερίδων φαγητού i που καταναλώνονται την ημέρα d
$y_{i,d}$: δυαδική μεταβλητή για την επιλογή της τροφής i την ημέρα d
$y_{i,d} = 1$, αν $x_{i,d} > 0$
$y_{i,d} = 0$, αν $x_{i,d} = 0$

Αντικειμενική συνάρτηση $\min z = \sum_{d=1}^D \sum_{i=1}^M c_i x_{i,d}$

Περιορισμοί
Για κάθε συστατικό n $N_n^{min} < \sum_{i=1}^M a_{ni} x_{i,d} < N_n^{max}$
$x_{i,d} \geq 0 \forall i, d$
$x_{i,d} \leq S y_{i,d} \forall i, d$ αν $y_{i,d} = 0$ τότε $x_{i,d} = 0$ (μη επιλεγμένη τροφή την ημέρα d)
$x_{i,d} \geq y_{i,d} \forall i, d$ αν $y_{i,d} = 1$ τότε $1 \leq x_{i,d} \leq S$ (επιλεγμένη τροφή με τουλάχιστον μία μερίδα)
$\sum_{d \in D} y_{i,d} \leq 1 \forall i$ κάθε φαγητό επιτρέπεται το πολύ σε μία μέρα

Πρόκειται για ένα μεικτό ακέραιο γραμμικό πρόβλημα με συνεχείς $x_{i,d}$ και δυαδικές μεταβλητές $y_{i,d}$ (MILP). Αποτελείται από $2 * M * D$ μεταβλητές,

$N * D$ περιορισμούς για τα θρεπτικά ορία, M περιορισμούς για την ποικιλία,

$2 * M * D$ περιορισμούς για τη σύνδεση των μεταβλητών x, y .

Λόγω των δυαδικών μεταβλητών η πολυπλοκότητα είναι NP-hard $O(2^{MD})$

G. Σενάριο 2 – Υλοποίηση

Σε αυτό το σενάριο έχουμε πάλι τις ίδιες παραμέτρους για φαγητά και συστατικά, $M=64$, $N=8$ και τα δεδομένα των πινάκων. Λύνουμε ένα ενιαίο πρόβλημα για όλο το σύνολο των ημερών με περιορισμό ποικιλίας, κάθε φαγητό να εμφανίζεται το πολύ σε μία μέρα


```

prob = pulp.LpProblem("full_diet", pulp.LpMinimize)

# Decision variables
x = pulp.LpVariable.dicts("x", ((foods[i], d) for i in range(M) for d in D), lowBound=0, cat='Continuous') #servings of food i in day d
y = pulp.LpVariable.dicts("y", ((foods[i], d) for i in range(M) for d in D), cat='Binary') #indicating if food i in day d is selected

# minimize cost
prob += pulp.lpSum(costs[i] * x[(foods[i], d)] for i in range(M) for d in D)

# Nutrition constraints
##for each nutrient and each day the total nutrient intake of the foods selected must be in [Nmin[n],Nmax[n]]
for n in range(N):
    for d in D:
        prob += pulp.lpSum(A[n][i] * x[(foods[i], d)] for i in range(M)) >= Nmin[n], f"min_nutrient_{n}_day_{d}"
        prob += pulp.lpSum(A[n][i] * x[(foods[i], d)] for i in range(M)) <= Nmax[n], f"max_nutrient_{n}_day_{d}"

##if y[f,d]=1 then 1<=x[f,d]<=S
for i in range(M):
    f = foods[i]
    for d in D:
        prob += x[(f, d)] <= S * y[(f, d)], f"link_upper_{f}_{d}"
        prob += x[(f, d)] >= y[(f, d)], f"link_lower_{f}_{d}"

# variety in foods
# each food is chosen once in the plan y[i,0]+y[i,1]+...<=1
for i in range(M):
    f = foods[i]
    prob += pulp.lpSum(y[(f, d)] for d in D) <= 1

# Solve the problem
prob.solve()

```

Figure 8: Μοντελοποίηση του προβλήματος σε κώδικα

Με τον solver CBC το πρόβλημα αυτό είναι πολύ βαρύ υπολογιστικά και απαιτεί πολύ χρόνο για να φτάσει στη βέλτιστη λύση. Για αυτό το λόγο οι μερίδες $x_{i,d}$ έγιναν **συνεχείς** μεταβλητές και παρακάτω θα δούμε το αποτέλεσμα για τιμές παραμέτρων $S=2$, $D=3$.

```

Status: Optimal
Objective: 50.330134716

Day 1:
Lettuce,Iceberg,Raw: 2.0 servings
Spaghetti W/ Sauce: 2.0 servings
Kiwifruit,Raw,Fresh: 1.1439221 servings
Oranges: 2.0 servings
Chocolate Chip Cookies: 1.180989 servings
Butter,Regular: 1.3255505 servings
3.3% Fat,Whole Milk: 1.160571 servings
Poached Eggs: 2.0 servings
Raisin Brn, Kellg's: 1.2064506 servings
Popcorn,Air-Popped: 2.0 servings
Tortilla Chip: 2.0 servings

Day 2:
Frozen Broccoli: 2.0 servings
Tofu: 2.0 servings
Tomato,Red,Ripe,Raw: 1.28423 servings
Apple,Raw,W/Skin: 2.0 servings
Banana: 2.0 servings
Wheat Bread: 2.0 servings
Oatmeal Cookies: 2.0 servings
Scrambled Eggs: 1.8676457 servings
Potato Chips,Bbqflvr: 2.0 servings
Pretzels: 1.3356422 servings
Splt Pea&Hamsoup: 2.0 servings

```

```

Day 3:
Carrots,Raw: 2.0 servings
Celery, Raw: 2.0 servings
Frozen Corn: 2.0 servings
Potatoes, Baked: 2.0 servings
White Bread: 2.0 servings
Skim Milk: 1.0356255 servings
Cheerios: 1.9489522 servings
Peanut Butter: 2.0 servings
Vegetbeef Soup: 1.8755239 servings
Beanbacn Soup,W/Watr: 2.0 servings

```

Figure 9: Βέλτιστη λύση $S=2$, $D=3$

	Day 1	Day 2	Day 3
Total Cost	15.11 €	17.25 €	17.97 €
Foods/Servings	Lettuce,Iceberg,Raw: 2.0 Spaghetti W/ Sauce: 2.0 KiwiFruit,Raw,Fresh: 1.1439221 Oranges: 2.0 Chocolate Chip Cookies: 1.180989 Butter,Regular: 1.3255505 3.3% Fat,Whole Milk: 1.160571 Poached Eggs: 2.0 Raisin Brn, Kellg'S: 1.2064506 Popcorn,Air-Popped: 2.0 Tortilla Chip: 2.0	Carrots,Raw: 2.0 Celery, Raw: 2.0 Frozen Corn: 2.0 Potatoes, Baked: 2.0 White Bread: 2.0 Skim Milk: 1.0356255 Cheerios: 1.9489522 Peanut Butter: 2.0 Vegetbeef Soup: 1.8755239 Beanbacn Soup,W/Watr: 2.0	Frozen Broccoli: 2.0 Tofu: 2.0 Tomato,Red,Ripe,Raw: 1.28423 Apple,Raw,W/Skin: 2.0 Banana: 2.0 Wheat Bread: 2.0 Oatmeal Cookies: 2.0 Scrambled Eggs: 1.8676457 Potato Chips,Bbqflvr: 2.0 Pretzels: 1.3356422 Splt Pea&Hamsoup: 2.0

Figure 10: Total plan D=3, S=2

H. Σενάριο 2 - Σχολιασμός των αποτελεσμάτων και ανάλυση για τις παραμέτρους S, D

Ο solver χρησιμοποιεί Branch and Bound και σε κάθε κόμβο λύνει ένα LP πρόβλημα. Το δέντρο έχει χιλιάδες κόμβους και η αναζήτηση είναι πολύ αργή. Η βέλτιστη λύση βρέθηκε σε χρόνο 523 sec και προτείνεται μία διατροφή με μέσο κόστος 16.77\$/day. Στη λύση που προτείνεται καμία τροφή δεν επαναλαμβάνεται στις ημέρες, οπότε ο περιορισμός δεν παραβιάζεται. Η λύση φαίνεται λιγότερο ρεαλιστική λόγω των συνεχών μερίδων.

Για S=1, το πρόβλημα γίνεται ανέφικτο ενώ για S=3 η βέλτιστη λύση βρίσκεται πολύ πιο γρήγορα σε 41 sec και είναι οικονομικότερη 11.33\$/day, όμως έχουμε λιγότερα διαφορετικά τρόφιμα μέσα στην ημέρα.

Για D=4, αυξάνεται εκθετικά ο χρόνος επίλυσης γιατί προστίθενται 2M νέες μεταβλητές και N περιορισμοί. Το συγκεκριμένο πρόβλημα για S=2 και D=4 είναι ανέφικτο.

```
Status: Optimal
Objective: 33.993832772

Day 1:
Frozen Broccoli: 3.0 servings
Celery, Raw: 2.4979508 servings
Potatoes, Baked: 3.0 servings
Banana: 2.8833389 servings
Poached Eggs: 3.0 servings
Scrambled Eggs: 3.0 servings
Tortilla Chip: 2.9813441 servings

Day 2:
Carrots,Raw: 3.0 servings
Lettuce,Iceberg,Raw: 2.0334242 servings
Apple,Raw,W/Skin: 2.9871698 servings
Oranges: 3.0 servings
Wheat Bread: 3.0 servings
3.3% Fat,Whole Milk: 1.8263119 servings
Peanut Butter: 3.0 servings
Popcorn,Air-Popped: 3.0 servings
Pretzels: 1.263827 servings
```

```
Day 3:
Spaghetti W/ Sauce: 3.0 servings
White Bread: 3.0 servings
Oatmeal Cookies: 1.5360414 servings
Chocolate Chip Cookies: 3.0 servings
Butter,Regular: 1.5099627 servings
Skim Milk: 1.1468594 servings
Beanbacn Soup,W/Watr: 1.2765552 servings
```

Figure 11: Βέλτιστη λύση $S=3$, $D=3$

Χαλαρώνοντας τον περιορισμό έτσι ώστε κάθε φαγητό να επιτρέπεται το πολύ δύο φορές στο πρόγραμμα προκύπτει σε χρόνο 470 sec η παρακάτω βέλτιστη λύση για $D=4$ και $S=2$. Τις δύο πρώτες μέρες προτείνονται τα ίδια φαγητά γιατί ο solver βλέπει ότι μπορεί να ξανα επιλέξει το ίδιο φθηνό/θρεπτικό φαγητό και το ίδιο γίνεται τις δύο επόμενες ημέρες με άλλο σετ τροφίμων.

```
Status: Optimal
Objective: 44.98645825600001

Day 1:
Frozen Broccoli: 2.0 servings
Carrots,Raw: 2.0 servings
Celery, Raw: 2.0 servings
Potatoes, Baked: 2.0 servings
Apple,Raw,W/Skin: 1.4061629 servings
Wheat Bread: 2.0 servings
Oatmeal Cookies: 1.5607883 servings
Chocolate Chip Cookies: 2.0 servings
Poached Eggs: 2.0 servings
Scrambled Eggs: 2.0 servings
White Rice: 1.0 servings
Tortilla Chip: 2.0 servings
Beanbacn Soup,W/Watr: 1.0884563 servings
```

```
Day 2:
Frozen Broccoli: 2.0 servings
Carrots,Raw: 2.0 servings
Celery, Raw: 2.0 servings
Potatoes, Baked: 2.0 servings
Apple,Raw,W/Skin: 1.4061629 servings
Wheat Bread: 2.0 servings
Oatmeal Cookies: 1.5607883 servings
Chocolate Chip Cookies: 2.0 servings
Poached Eggs: 2.0 servings
Scrambled Eggs: 2.0 servings
White Rice: 1.0 servings
Tortilla Chip: 2.0 servings
Beanbacn Soup,W/Watr: 1.0884563 servings
```

```
Day 3:
Lettuce,Iceberg,Raw: 1.0 servings
Spaghetti W/ Sauce: 1.9913793 servings
Banana: 2.0 servings
Oranges: 2.0 servings
White Bread: 2.0 servings
3.3% Fat,Whole Milk: 1.5175979 servings
Peanut Butter: 2.0 servings
Popcorn,Air-Popped: 2.0 servings
```

```
Day 4:
Lettuce,Iceberg,Raw: 1.0 servings
Spaghetti W/ Sauce: 1.9913793 servings
Banana: 2.0 servings
Oranges: 2.0 servings
White Bread: 2.0 servings
3.3% Fat,Whole Milk: 1.5175979 servings
Peanut Butter: 2.0 servings
Popcorn,Air-Popped: 2.0 servings
```

Figure 12: Βέλτιστη λύση με χαλαρό περιορισμό $D=4$, $S=2$

I. Σύγκριση σεναρίων

Το πρώτο σενάριο δίνει γρήγορα μία λύση σχηματίζοντας υποπροβλήματα, αλλά τα τρόφιμα με χαμηλό κόστος/υψηλή αξία τείνουν να επαναλαμβάνονται.

Το δεύτερο σενάριο απαιτεί περισσότερο χρόνο να βρει λύση λόγω του ισχυρού περιορισμού ανάμεσα στις ημέρες και των δυαδικών μεταβλητών, αλλά είναι πιο ισορροπημένο και οι φθηνές/θρεπτικές επιλογές κατανέμονται σε όλες τις μέρες.

J. Βιβλιογραφία

1. <https://neos-guide.org/case-studies/om/the-diet-problem/>
2. [LPmodelDietProblem Hretcanu 2010.pdf](#)
3. https://eclass.upatras.gr/modules/document/file.php/EE916/%CE%92%CE%B9%CE%B2%CE%BB%CE%B9%CE%BF%CE%B3%CF%81%CE%B1%CF%86%CE%AF%CE%B1/Optimal%20Diet%20Problem/Stigler%27sDietProblemRevised_GarilleGass_2001.pdf