Επιχειρησιακή Έρευνα

Γιώργος Τσιρογιάννης

Τμήμα Διοίκησης Επιχειρήσεων Αγροτικών Προϊόντων και Τροφίμων,

Πανεπιστήμιο Πατρών

Ακαδημαϊκό έτος: 2022-2023

Βασικά στάδια της Επιχειρησιακής Έρευνας

- Ανάλυση του συστήματος
- Διατύπωση στόχων
- **Διατύπωση του μοντέλου**
- **Επίλυση του μοντέλου**
- Ανάλυση της λύσης
- Υλοποίηση της λύσης

Διατύπωση του μοντέλου

- **Ζητούμενο:** να δημιουργήσουμε μια απλουστευμένη αναπαράσταση του πραγματικού συστήματος, να τη μελετήσουμε, να αναλύσουμε την επίδραση διαφόρων παραγόντων στους στόχους που έχουν τεθεί και να επιλέξουμε την καλύτερη στρατηγική.
- **Μορφή:** μετατροπή του ορισμού του προβλήματος σε μαθηματικές σχέσεις και αποτελεί κατά προσέγγιση μία αναπαράσταση του προβλήματος. Συνήθως ένα σύνολο από ποσοτικές σχέσεις ή εντολές στον υπολογιστή που εκφράζουν τους στόχους του προβλήματος και τους περιορισμούς του περιβάλλοντος

Διατύπωση του μοντέλου (συν.)

- **Μεθοδολογία**
  - **Στάδιο 1ο:** Διατύπωση υποθέσεων οι οποίες απλουστεύουν το πρόβλημα.
    Σκοπιμότητα: Αυτό γίνεται για να κάνει πιο εφικτή και εύκολη την επίλυση και ανάλυση του προβλήματος.
  - **Στάδιο 2ο:** Διατύπωση εκείνων των μαθηματικών σχέσεων ή εντολών στον υπολογιστή, οι οποίες εκφράζουν τις σχέσεις μεταξύ των συντελεστών του συστήματος, των στόχων, των μεταβλητών και του περιβάλλοντος.
    Σκοπιμότητα: αυστηρότερη διατύπωση του μοντέλου.
  - **Στάδιο 3ο: Ε**πιβεβαίωση του μοντέλου με δοκιμαστική χρήση του σε ένα "απλό" πρόβλημα. Αυτό γίνεται για να ελέγξουμε την ακρίβεια των υποθέσεων (στάδιο 1) και των σχέσεων/εντολών (στάδιο 2) που διατυπώθηκαν παραπάνω. Σε περίπτωση που τα αποτελέσματα δεν είναι ικανοποιητικά, τα στάδια 1,2 επαναλαμβάνονται.

Διατύπωση του μοντέλου (συν.)

- **Κρισιμότητα:** ιδιαίτερα υψηλή
  - Ο αναλυτής ενός συστήματος διατρέχει τον κίνδυνο να εντοπίσει τη σωστή λύση σε λάθος πρόβλημα (το προτεινόμενο μοντέλο δεν αποτελεί σωστή αναπαράσταση του προβλήματος, ενώ η μέθοδος βελτιστοποίησης της αντικειμενικής συνάρτησης είναι σωστή). Στην αντίθετη περίπτωση κινδυνεύει να δώσει λάθος λύση σε σωστό πρόβλημα.
- **Δυσκολίες/προκλήσεις:** περιορισμός των μεταβλητών σε εκείνες που είναι πραγματικά σημαντικές για το πρόβλημα, δεδομένου ότι από αυτό εξαρτάται το κόστος επίλυσής του.

Επίλυση του μοντέλου

- **Ζητούμενο:** εντοπισμός της βέλτιστης λύσης του προβλήματος.
- **Τρόποι επίλυσης:** στηρίζονται σε Ανώτερα Μαθηματικά (διαφορικός και ολοκληρωτικός λογισμός, αριθμητική ανάλυση, γραμμική άλγεβρα, κλασικές μέθοδοι βελτιστοποίησης, λογισμός των μεταβολών), στη θεωρία Πιθανοτήτων και τη Στατιστική (θεωρία πιθανοτήτων, ανελίξεις Markov, περιγραφική στατιστική, στατιστική συμπερασματολογία, εκτίμηση, ανάλυση παλινδρόμησης, αλληλοσυσχέτιση, ανάλυση μεταβλητότητας, παραγοντική ανάλυση, χρονοσειρές) ή σε Μεθόδους και Θεωρίες της επιχειρησιακής έρευνας.

Διατύπωση του μοντέλου
Επίλυση του μοντέλου

Παραδείγματα

Η επιχειρησιακή έρευνα ασχολείται με την **άριστη λήψη αποφάσεων** σε **προσδιοριστικά και πιθανολογικά συστήματα** που προκύπτουν μέσα από **πραγματικά προβλήματα**.

- **Παραγωγή**: Χωροθέτηση εργοστασίου, προγραμματισμός προμηθειών ή παραγωγής, έλεγχος αποθεμάτων πρώτων υλών ή προϊόντων, έλεγχος ποιότητας, ανανέωση μηχανολογικού εξοπλισμού, ανάλυση αξιοπιστίας, εξισορρόπηση γραμμής παραγωγής.
- **Προσωπικό:** Ανάλυση αξιολόγηση αξιολόγηση προσωπικού, προγραμματισμός προσωπικού, ανάλυση αιτιών απουσίας, πρόληψη ατυχημάτων.

Η επιχειρησιακή έρευνα ασχολείται με την **άριστη λήψη αποφάσεων** σε **προσδιοριστικά και πιθανολογικά συστήματα** που προκύπτουν μέσα από **πραγματικά προβλήματα**.

- **Εμπόριο:** Καθορισμός βέλτιστης σύνθεσης παραγωγής, βέλτιστη στρατηγική διαφημίσεως και τιμολογήσεως προϊόντων, προγραμματισμός πωλήσεων, προγραμματισμός μεταφοράς και διανομής προϊόντων, προσδιορισμός θέσεως και αριθμού αποθηκών, σύνθεση μεταφορικού στόλου.
- **Οικονομικά:** Χρηματοοικονομικός προγραμματισμός, καθορισμός πιστωτικής πολιτικής, προϋπολογισμός, βελτιστοποίηση χρηματοροών.

Παραδείγματα εφαρμογών

Άριστη λήψη αποφάσεων σε συστήματα που προκύπτουν από πραγματικά προβλήματα

Παραγωγή

Παραδείγματα

Παράδειγμα 1

- Έστω βιομηχανική μονάδα
- Με 3 μηχανές M_1, M_2, M_3
- Παραγωγή 4 ειδών προϊόντων $\Pi_1, \Pi_2, \Pi_3, \Pi_4$
- Ο χρόνος επεξεργασίας ανά προϊόν (σε λεπτά-min), ο διαθέσιμος ημερήσιος χρόνος ανά μηχανή και το κέρδος όπως έχουν προκύψει από την ανάλυση συστήματος.

	Χρόνος επεξεργασίας κάθε προϊόντος στις διάφορες μηχανές				Διαθέσιμος ημερήσιος χρόνος κάθε μηχανής (min)
	Προϊόν 1	Προϊόν 2	Προϊόν 3	Προϊόν 4	
Μηχανή 1	4	2	3	1	480
Μηχανή 2	2	3	1	3	360
Μηχανή 3	3	0	1	0	240
Κέρδος ανά μονάδα προϊόντος	6	4	3	5	

Ποια στρατηγική επιτυγχάνει το μέγιστο κέρδος;

*: 0 (μηδέν) χρόνος σημαίνει ότι η μηχανή δεν μπορεί να παράγει το προϊόν αυτό.

Παράδειγμα 1ο (συν.)

Εντοπίζουμε τις σταθερές/παραμέτρους του προβλήματος

	Χρόνος επεξεργασίας κάθε προϊόντος στις διάφορες μηχανές				Διαθέσιμος ημερήσιος χρόνος κάθε μηχανής (min)
	Προϊόν 1	Προϊόν 2	Προϊόν 3	Προϊόν 4	
Μηχανή 1	4	2	3	1	480
Μηχανή 2	2	3	1	3	360
Μηχανή 3	3	0	1	0	240
Κέρδος ανά μονάδα προϊόντος	6	4	3	5	

Παράδειγμα 1ο (συν.)

Διατυπώνουμε τις μεταβλητές (μεταβλ. αποφάσεων)

- Οι ποσότητες x_1, x_2, x_3, x_4 (οι άγνωστοι του προβλήματος) από τα προϊόντα $\Pi_1, \Pi_2, \Pi_3, \Pi_4$ που πρέπει να παραχθούν ώστε να μεγιστοποιηθεί το κέρδος.

Παράδειγμα 1ο (συν.)

Διατύπωση του κριτηρίου και της αντικειμενικής συνάρτησης

- Επειδή πρόκειται για κέρδος → μεγιστοποίηση
- Αντικειμενική συνάρτηση:

$$z = 6 * x_1 + 4 * x_2 + 3 * x_3 + 5 * x_4$$

Παράδειγμα 1 (συν.)

Διατύπωση των περιορισμών

- Χωρίς τους περιορισμούς → όσο μεγαλύτερες ποσότητες τόσο μεγαλύτερο κέρδος.
- Πρέπει να ληφθούν υπόψη οι χρονικοί περιορισμοί

Παράδειγμα 1ο (συν.)

Διατύπωση των περιορισμών

	Χρόνος επεξεργασίας κάθε προϊόντος στις διάφορες μηχανές				Διαθέσιμος ημερήσιος χρόνος κάθε μηχανής (min)
	Προϊόν 1	Προϊόν 2	Προϊόν 3	Προϊόν 4	
Μηχανή 1	4	2	3	1	480
Μηχανή 2	2	3	1	3	360
Μηχανή 3	3	0	1	0	240
Κέρδος ανά μονάδα προϊόντος	6	4	3	5	

$$\left.\begin{array}{rcrcrcrcr} 4x_1 & + & 2x_2 & + & 3x_3 & + & x_4 & \leq & 480 \\ 2x_1 & + & 3x_2 & + & x_3 & + & 3x_4 & \leq & 360 \\ 3x_1 & + & & + & x_3 & & & \leq & 240 \end{array}\right\}$$

Περιορισμοί διαθεσιμότητας του ημερήσιου χρόνου των μηχανών

$$x_1, x_2, x_3, x_4 \geq 0$$

Περιορισμοί μη αρνητικότητας

Παράδειγμα 1ο (συν.)

Πλήρες μαθηματικό μοντέλο

Αντικειμενική συνάρτηση:

$$\text{maximize Profit} = \max z = 6x_1 + 4x_2 + 3x_3 + 5x_4$$ Μεγιστοποίηση κέρδους

Υπό των περιορισμών:

$$\left.\begin{array}{ccccccccc} 4x_1 & + & 2x_2 & + & 3x_3 & + & x_4 & \leq & 480 \\ 2x_1 & + & 3x_2 & + & x_3 & + & 3x_4 & \leq & 360 \\ 3x_1 & + & & + & x_3 & & & \leq & 240 \end{array}\right\}$$

Περιορισμοί διαθεσιμότητας του ημερήσιου χρόνου των μηχανών

$$x_1, x_2, x_3, x_4 \geq 0$$ Περιορισμοί μη αρνητικότητας

Παράδειγμα 1ο (συν.)

Επίλυση στον υπολογιστή


- ```
  value(x1) = 60.0
  value(x2) = 0.0
  value(x3) = 59.999
  value(x4) = 60.0
  objective_value(model) = 780.0
  ```
  ```

Οικονομικά

Παραδείγματα

Παράδειγμα 2ο

- Εταιρεία προετοιμάζει επένδυση έως 100Μ ευρώ.
- Περιοχές επένδυσης: κοινές μετοχές (ΚΜ), προνομιούχες μετοχές(ΠΜ), ομόλογα δημοσίου(ΟΔ).
- Αναμενόμενες αποδόσεις: ΚΜ:15%, ΠΜ:10%, ΟΔ:7%
- Λόγω ρίσκου επένδυσης υπάρχουν περιορισμοί σε σχέση με την συνολικής επένδυσης (ΣΕ): ΚΜ όχι παραπάνω από 25% του ΣΕ, ΠΜ το πολύ 50% συνολικού ποσού μετοχών, τουλάχιστον 20% της ΣΕ σε ΟΔ
- Ποια στρατηγική μεγιστοποιεί το κέρδος

Παράδειγμα 2 (συν.)

Διατυπώνουμε τις μεταβλητές (μεταβλ. αποφάσεων)

- Οι ποσότητες x_1, x_2, x_3 (οι άγνωστοι του προβλήματος σε εκατομμύρια) από τις περιοχές επένδυσης ΚΜ, ΠΜ, ΟΔ για την μεγιστοποιεί το κέρδος

Παράδειγμα 2ο (συν.)

Διατύπωση του κριτηρίου και της αντικειμενικής συνάρτησης

- Επειδή πρόκειται για κέρδος → μεγιστοποίηση
- Αντικειμενική συνάρτηση:

$$z = 0.15 * x_1 + 0.10 * x_2 + 0.07 * x_3$$

Παράδειγμα 2ο (συν.)

Διατύπωση των περιορισμών

Λόγω ρίσκου επένδυσης υπάρχουν περιορισμοί σε σχέση με την συνολικής επένδυσης (ΣΕ): ΚΜ όχι παραπάνω από 25% του ΣΕ, ΠΜ το πολύ 50% συνολικού ποσού μετοχών, τουλάχιστον 20% της ΣΕ σε ΟΔ

$$\text{ΣΕ} = \text{ΚΜ} + \text{ΠΜ} + \text{ΟΔ} = x_1 + x_2 + x_3$$

$$x_1 \leq 0.25 * (x_1 + x_2 + x_3) \rightarrow 0.75 * x_1 - 0.25 * x_2 - 0.25 * x_3 \leq 0$$

$$x_2 \leq 0.50 * (x_1 + x_2) \rightarrow -0.50 * x_1 + 0.50 * x_2 \leq 0$$

$$x_3 \geq 0.20 * (x_1 + x_2 + x_3) \rightarrow -0.20 * x_1 - 0.20 * x_2 + 0.80 * x_3 \geq 0$$

Παράδειγμα 2ο (συν.)

Διατύπωση των περιορισμών

Εταιρεία προετοιμάζει επένδυση έως 100Μ ευρώ.

$$\Sigma\text{Ε} = \text{ΚΜ} + \text{ΠΜ} + \text{ΟΔ} = x_1 + x_2 + x_3$$

$$x_1 + x_2 + x_3 \leq 100$$

$$x_1 \geq 0$$

$$x_2 \geq 0$$

$$x_3 \geq 0$$

Παράδειγμα 2ο (συν.)

Πλήρες μαθηματικό μοντέλο

Αντικειμενική συνάρτηση:

$$\max z = 0.15 * x_1 + 0.10 * x_2 + 0.07 * x_3$$

Υπό των περιορισμών:

$$0.75 * x_1 - 0.25 * x_2 - 0.25 * x_3 \leq 0$$

$$-0.50 * x_1 + 0.50 * x_2 \leq 0$$

$$-0.20 * x_1 - 0.20 * x_2 + 0.80 * x_3 \geq 0$$

$$x_1 + x_2 + x_3 \leq 100$$

$$x_1 \geq 0$$

$$x_2 \geq 0$$

$$x_3 \geq 0$$

Παράδειγμα 2ο (συν.)

Επίλυση στον υπολογιστή


```
value(x1) = 25.0

value(x2) = 25.0

value(x3) = 50.0

objective_value(model) = 9.75
```

Παραγωγή

Παραδείγματα

Παράδειγμα 3ο

- Βιομηχανία ζωοτροφών.
- Το προϊόν πρέπει να περιέχει τουλάχιστον 25% πρωτεΐνη και λιγότερο από 10% λιπαρά
- Ποια είναι η οικονομικότερη στρατηγική;

Θρεπτικά Συστατικά	**Περιεκτικότητα (%) των πρώτων υλών σε θρεπτικά συστατικά**				**Επιθυμητή περιεκτικότητα (%)**
	Κριθάρι	**Βρώμη**	**Σουσάμι**	**Καλαμποκάλευρο**	
Πρωτεΐνες	15	12	40	50	τουλάχιστον 25
Λιπαρά	4	8	15	5	το πολύ 10
Κόστος πρώτων υλών (€ ανά τόνο)	800	950	1400	1200	

Παράδειγμα 3ο (συν.)

Διατυπώνουμε τις μεταβλητές (μεταβλ. αποφάσεων)

- Οι ποσότητες x_1, x_2, x_3, x_4 αντιπροσωπεύουν τις ποσότητες από Κριθάρι (Κρ), Βρώμη (Βρ), Σουσάμι (Σο) και Καλαμπόκι (Κα) αντίστοιχα.
- Επειδή το κόστος δίνεται ανά τόνο, θα υπολογίσουμε και τις ποσότητες ώστε το σύνολο να είναι ένας τόνος.

Παράδειγμα 3ο (συν.)

Διατύπωση του κριτηρίου και της αντικειμενικής συνάρτησης

- Επειδή πρόκειται για κόστος → ελαχιστοποίηση
- Αντικειμενική συνάρτηση:

$$z = 800 * x_1 + 950 * x_2 + 1400 * x_3 + 1200 * x_4$$

Παράδειγμα 3ο (συν.)

Διατύπωση των περιορισμών

Ποσότητες ανά τόνο: $x_1 + x_2 + x_3 + x_4 = 1$

Περιορισμοί από προδιαγραφές προϊόντος:

$$\frac{0.15 * x_1 + 0.12 * x_2 + 0.40 * x_3 + 0.50 * x_4}{x_1 + x_2 + x_3 + x_4} \geq 0.25$$

$$\frac{0.04 * x_1 + 0.08 * x_2 + 0.15 * x_3 + 0.05 * x_4}{x_1 + x_2 + x_3 + x_4} \leq 0.10$$

$$x_1 \geq 0$$

$$x_2 \geq 0$$

$$x_3 \geq 0$$

$$x_4 \geq 0$$

Παράδειγμα 3ο (συν.)

Πλήρες μαθηματικό μοντέλο

Αντικειμενική συνάρτηση:

$$\min z = 800 * x_1 + 950 * x_2 + 1400 * x_3 + 1200 * x_4$$

Υπό των περιορισμών:

$$x_1 + x_2 + x_3 + x_4 = 1$$

$$0.15 * x_1 + 0.12 * x_2 + 0.40 * x_3 + 0.50 * x_4 \geq 0.25$$

$$0.04 * x_1 + 0.08 * x_2 + 0.15 * x_3 + 0.05 * x_4 \leq 0.10$$

$$x_1 \geq 0$$

$$x_2 \geq 0$$

$$x_3 \geq 0$$

$$x_4 \geq 0$$

Παράδειγμα 3ο (συν.)

Επίλυση στον υπολογιστή


```
value(x1) = 0.7142857142857143

value(x2) = 0.0

value(x3) = 0.0

value(x4) = 0.2857142857142857

objective_value(model) = 914.2857142857142
```

Εμπόριο

Παραδείγματα

Παράδειγμα 4ο

- Βιομηχανία εμφιάλωσης νερού.
- 3 εργοστάσια (Έδεσσα, Λουτράκι, Λ. Όρη)
- 4 διαμετακομιστικό κέντρα (Ηράκλειο, Κερατσίνι, Βόλος, Θεσσαλονίκη)
- Στήλη προσφοράς: η μέγιστη δυνατότητα παραγωγής σε χιλιάδες κιβώτια
- Γραμμή ζήτησης: η ζήτηση σε χιλιάδες κιβώτια
- Κόστος μεταφοράς: ευρώ/χίλια κιβώτια
- Ποια είναι η οικονομικότερη στρατηγική;

		ΠΡΟΟΡΙΣΜΟΣ				ΠΡΟΣΦΟΡΑ
		Ηράκλειο	Κερ/νι	Βόλος	Θεσ/νίκη	
ΠΡΟΕΛΕΥΣΗ	Έδεσσα	540	420	340	160	400
	Λουτράκι	180	80	240	300	850
	Λ. Όρη	120	300	500	440	350
ΖΗΤΗΣΗ		150	700	250	500	**1600**

Παράδειγμα 4ο (συν.)

Διατυπώνουμε τις μεταβλητές (μεταβλ. αποφάσεων)

- Οι μεταβλητές x_i αντιπροσωπεύουν τις ποσότητες που παράγονται και μεταφέρονται στα διαμετακομιστικό κέντρα (σε χιλιάδες κιβώτια)

		ΠΡΟΟΡΙΣΜΟΣ			
		Ηράκλειο	**Κερ/νι**	**Βόλος**	**Θεσ/νίκη**
ΠΡΟΕΛΕΥΣΗ	**Έδεσσα**	x_1	x_2	x_3	x_4
	Λουτράκι	x_5	x_6	x_7	x_8
	Λ. Όρη	x_9	x_{10}	x_{11}	x_{12}

Παράδειγμα 4ο (συν.)

		ΠΡΟΟΡΙΣΜΟΣ			
		Ηράκλειο	Κερ/νι	Βόλος	Θεσ/νίκη
ΠΡΟΕΛΕΥΣΗ	Έδεσσα	540	420	340	160
	Λουτράκι	180	80	240	300
	Λ. Όρη	120	300	500	440

Διατύπωση του κριτηρίου και της αντικειμενικής συνάρτησης

- Επειδή πρόκειται για κόστος → ελαχιστοποίηση
- Αντικειμενική συνάρτηση:

$$z = 540 * x_1 + 420 * x_2 + 340 * x_3 + 160 * x_4 + 180 * x_5 + 80 * x_6 + 240 * x_7 + 300 * x_8 + 120 * x_9 + 300 * x_{10} + 500 * x_{11} + 440 * x_{12}$$

Παράδειγμα 4ο (συν.)

Διατύπωση των περιορισμών

$$x_1 + x_2 + x_3 + x_4 \leq 400$$

$$x_5 + x_6 + x_7 + x_8 \leq 850$$

$$x_9 + x_{10} + x_{11} + x_{12} \leq 350$$

		ΠΡΟΟΡΙΣΜΟΣ				ΠΡΟΣΦΟΡΑ
		Ηράκλειο	Κερ/νι	Βόλος	Θεσ/νίκη	
ΠΡΟΕΛΕΥΣΗ	Έδεσσα	x_1	x_2	x_3	x_4	400
	Λουτράκι	x_5	x_6	x_7	x_8	850
	Λ. Όρη	x_9	x_{10}	x_{11}	x_{12}	350
ΖΗΤΗΣΗ		150	700	250	500	1600

$$x_1 + x_5 + x_9 = 150$$

$$x_2 + x_6 + x_{10} = 700$$

$$x_3 + x_7 + x_{11} = 250$$

$$x_4 + x_8 + x_{12} = 500$$

$$x_i \geq 0, i = 1 \dots 12$$

Παράδειγμα 4ο (συν.)

Πλήρες μαθηματικό μοντέλο

Αντικειμενική συνάρτηση:

$$\min z = 540 * x_1 + 420 * x_2 + 340 * x_3 + 160 * x_4 + 180 * x_5 + 80 * x_6 + 240 * x_7 + 300 * x_8 + 120 * x_9 + 300 * x_{10} + 500 * x_{11} + 440 * x_{12}$$

Υπό των περιορισμών:

$$x_1 + x_2 + x_3 + x_4 \le 400$$

$$x_5 + x_6 + x_7 + x_8 \le 850$$

$$x_9 + x_{10} + x_{11} + x_{12} \le 350$$

$$x_1 + x_5 + x_9 = 150$$

$$x_2 + x_6 + x_{10} = 700$$

$$x_3 + x_7 + x_{11} = 250$$

$$x_4 + x_8 + x_{12} = 500$$

$$x_ì \ge 0, i = 1 \ldots 12$$

Παράδειγμα 4ο (συν.)

Επίλυση στον υπολογιστή

```
value(x1) = 0.0

value(x2) = 0.0

value(x3) = 0.0

value(x4) = 480.0

value(x5) = 0.0

value(x6) = 600.0

value(x7) = 250.0

value(x8) = 0.0

value(x9) = 150.0

value(x10) = 100.0

value(x11) = 0.0

value(x12) = 20.0

objective_value(model) = 241600.0
```

Βιομηχανία

Παραδείγματα

Παράδειγμα 5ο

- Βιομηχανία παραγωγής πετρελαιοειδών.
- Δύο τύποι βενζίνης συγκεκριμένων προδιαγραφών και ποσοτήτων.

Τύποι βενζίνης	Βαθμός οκτανίων	Πίεση αερίων	Διαθέσιμη ποσότητα (σε βαρέλια)
Τύπος 1	105	5	40,000
Τύπος 2	95	10	100,000

- Από ανάμιξη προκύπτει: Καύσιμο αεροπ. τύπου, καύσιμο αυτοκινήτων.
- Οι δύο τύποι καυσίμων είναι συγκεκριμένων προδιαγραφών και τιμών.

Τελικό προϊόν	Ελάχιστος Βαθμός οκτανίων	Μέγιστη πίεση αερίων	Μέγιστες πωλήσεις (σε βαρέλια)	Τιμή πώλησης (€ / βαρέλι)
Καύσιμο αεροπορικού τύπου	104	7	35,000	150
Καύσιμο αυτοκίνησης	96	9	οποιαδήποτε ποσότητα	110

- Στόχος: βέλτιστη στρατηγική για τις διαθέσιμες ποσότητες ΄(τιμή πώλησης = κόστος).

Παράδειγμα 5ο (συν.)

Διατυπώνουμε τις μεταβλητές (μεταβλ. αποφάσεων)

- Η μεταβλητή x_1, x_2 αντιπροσωπεύει τα βαρέλια του καυσίμου ΑΤ (αεροπορικού τύπου), ενώ η x_3, x_4 αντιπροσωπεύει τα βαρέλια του καυσίμου ΑΥ (αυτοκινήτου) από την βενζίνη Τ1, Τ2 αντίστοιχα.

Παράδειγμα 5ο (συν.)

Διατύπωση του κριτηρίου και της αντικειμενικής συνάρτησης

- Επειδή πρόκειται για κόστος → ελαχιστοποίηση
- Αντικειμενική συνάρτηση:

$$z = 150 * (\mathrm{x}_1 + \mathrm{x}_2) + 110 \ * (\mathrm{x}_3 + \mathrm{x}_4)$$

Παράδειγμα 5ο (συν.)

Διατύπωση των περιορισμών

- Περιορισμοί ποσότητας:

$$x_1 + x_3 \leq 40000 \qquad x_2 + x_4 \leq 100000$$

- Περιορισμοί οκτανίων:

$$105 * x_1 + 95 * x_2 \geq 104 * (x_1 + x_2) \qquad 105 * x_3 + 95 * x_4 \geq 96 * (x_3 + x_4)$$

- Περιορισμοί πίεσης:

$$5 * x_1 + 10 * x_2 \leq 7 * (x_1 + x_2) \qquad 5 * x_3 + 10 * x_4 \leq 9 * (x_3 + x_4)$$

- Περιορισμός ποσότητας πώλησης:

$$x_1 + x_2 \leq 35000$$

- Περιορισμοί θετικότητας:

$$x_1 \geq 0 \qquad x_2 \geq 0 \qquad x_3 \geq 0 \qquad x_4 \geq 0$$

Παράρτημα

Οδηγίες επίλυσης σε Julia

The Julia Programming Language

julialang.org

Get a limited edition Julia Language Hacktoberfest shirt on Bonfire

julia

Download Documentation Blog Community Learn Research JSoC

Sponsor

The Julia Programming Language

Download v1.5.2 | Documentation

Star 29,755

Julia in a Nutshell

Fast

Julia was designed from the beginning for high performance. Julia programs compile to efficient native code for multiple platforms via LLVM.

Dynamic

Julia is dynamically typed, feels like a scripting language, and has good support for interactive use.

Reproducible

Reproducible environments make it possible to recreate the same Julia environment every time, across platforms, with pre-built binaries.

Composable

Julia uses multiple dispatch as a paradigm, making it easy to express many object-oriented and functional programming patterns. The talk on the Unreasonable Effectiveness of Multiple Dispatch explains why it works so well.

General

Julia provides asynchronous I/O, metaprogramming, debugging, logging, profiling, a package manager, and more. One can build entire Applications and Microservices in Julia.

Open source

Julia is an open source project with over 1,000 contributors. It is made available under the MIT license. The source code is available on GitHub.

See Julia Code Examples | Try Julia In Your Browser

The Julia Programming Language | JuliaPro – Julia Computing

juliacomputing.com/products/juliapro.html

Julia computing

PRODUCTS INDUSTRIES CASE STUDIES EVENTS TRAINING MEDIA RESOURCES BLOG

JP Julia PRO

Seamless Development

Version : 1.5.2-1

AMI Links | FAQs | Documentation

FREE DOWNLOAD

JuliaPro is free to download and is the fastest on-ramp to Julia for individual researchers, engineers, scientists, quants, traders, economists, students and others. Julia developers can build better software quicker and easier while benefiting from Julia's unparalleled high performance.

JuliaPro is lightweight and easy to install. Use any package from 2600+ open source packages or from a curated list of 250+ JuliaPro packages. Curated packages are tested, documented and supported by Julia Computing. See below for details on curated packages.

Current stable release (v1.5.2-1)

Operating system	v1.5.2-1*	Quick-start Guide
Windows		
Mac**		
Linux		
Linux (GPG)		

Release with long-term support (v1.0.5-2)

Operating system	v1.0.5-2*	Quick-start Guide
Windows		
Mac**		
Linux		
Linux (GPG)		

OUR OTHER ENTERPRISE PRODUCTS

Julia SURE

Julia TEAM

Julia RUN

COMPARE FEATURES

The Julia Programming Language × JuliaPro – Julia Computing × Think Julia: How to Think Like a C ×

benlauwens.github.io/ThinkJulia.jl/latest/book.html

Table of Contents

Think Julia: How to Think Like a Computer Scientist

Ben Lauwens – ben.lauwens@gmail.com · Allen Downey – allendowney@gmail.com

Copyright

Ben Lauwens is a Professor of Mathematics at Royal Military Academy (RMA Belgium). He has a PhD in Engineering and Master's degrees from KU Leuven and RMA and Bachelor's degree from RMA.

Allen Downey is a Professor of Computer Science at Olin College of Engineering. He has taught at Wellesley College, Colby College and U.C. Berkeley. He has a PhD in Computer Science from U.C. Berkeley and Master's and Bachelor's degrees from MIT.

A paper version of this book is published by O'Reilly Media: http://shop.oreilly.com/product/0636920215707.do and can be bought on Amazon: https://www.amazon.com/Think-Julia-Like-Computer-Scientist/dp/1492045039.

Dedication

For Emeline, Arnaud and Tibo.

Preface

In January 2018 I started the preparation of a programming course targeting students without programming experience. I

The Julia Programming Language | JuliaPro – Julia Computing | Think Julia: How to Think Like a C | Introduction · JuMP

jump.dev/JuMP.jl/stable/

JuMP

v0.21.5

Search docs

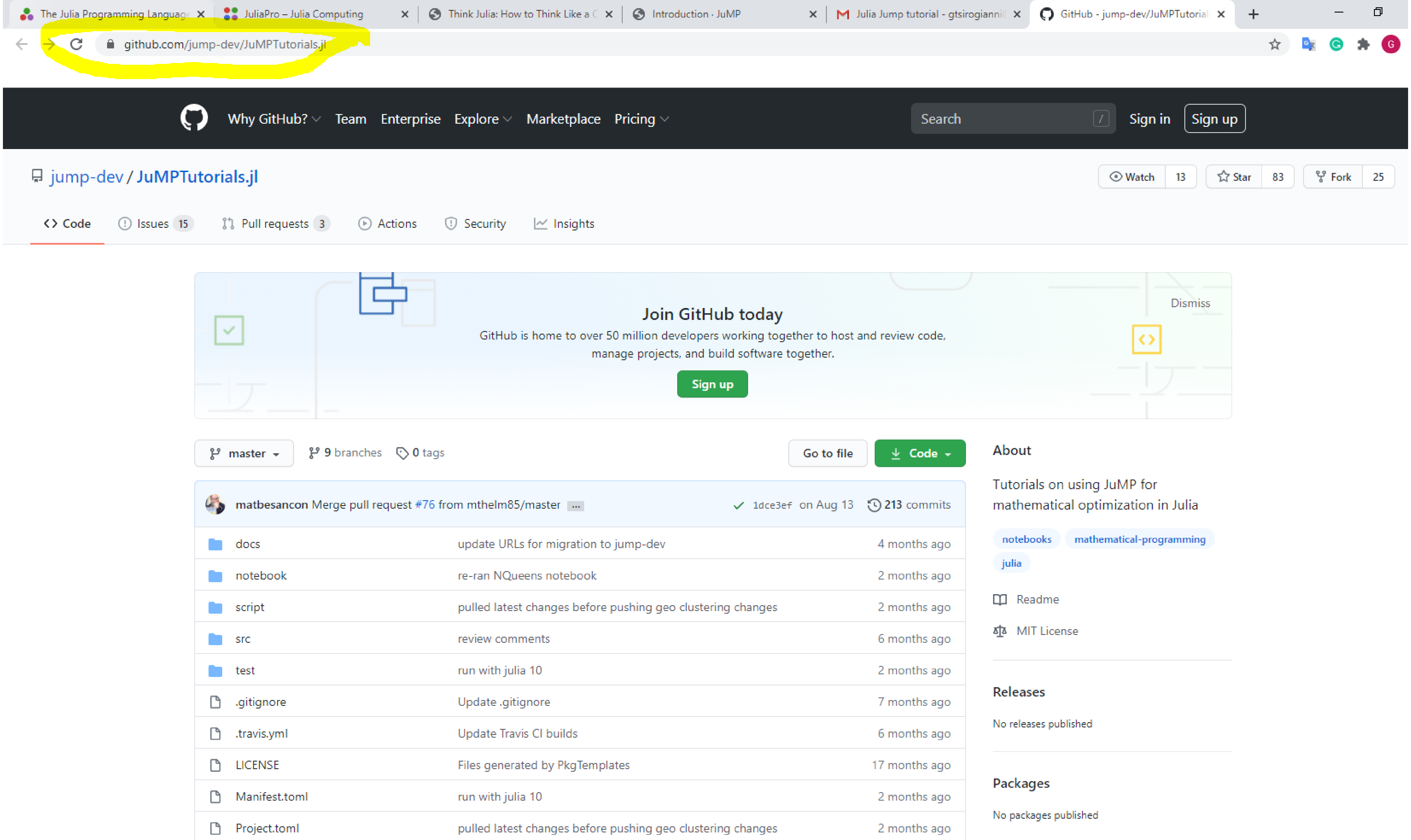
powered by NumFOCUS

Warning

Between versions 0.18 and 0.19, JuMP underwent a major transition in its underlying solver abstraction API, from MathProgBase to MathOptInterface. See NEWS.md for a comprehensive list of changes between the two versions, many of which are breaking. This documentation is for JuMP/MathOptInterface. For the documentation of JuMP 0.18, see here.

JuMP is a domain-specific modeling language for mathematical optimization embedded in Julia. It currently supports a number of open-source and commercial solvers (see below) for a variety of problem classes, including **linear programming**, **mixed-integer programming**, **second-order conic programming**, **semidefinite programming**, and **nonlinear programming**. JuMP's features include:

- User friendliness
  - Syntax that mimics natural mathematical expressions.
  - Complete documentation (WIP!)
- Speed
  - Benchmarking has shown that JuMP can create problems at similar speeds to special-purpose modeling languages such as AMPL.
  - JuMP communicates with most solvers in memory, avoiding the need to write intermediary files.
- Solver independence
  - JuMP uses a generic solver-independent interface provided by the MathOptInterface package,


The Julia Programming Language
JuliaPro – Julia Computing
Think Julia: How to Think Like a C
Introduction · JuMP
Julia Jump tutorial - gtsirogianni
GitHub - jump-dev/JuMPTutorial
github.com/jump-dev/JuMPTutorials.jl
Why GitHub?
Team
Enterprise
Explore
Marketplace
Pricing
Search
Sign in
Sign up
jump-dev / JuMPTutorials.jl
Watch 13
Star 83
Fork 25
Code
Issues 15
Pull requests 3
Actions
Security
Insights
Join GitHub today
GitHub is home to over 50 million developers working together to host and review code, manage projects, and build software together.
Sign up
Dismiss
master
9 branches
0 tags
Go to file
Code
matbesancon Merge pull request #76 from mthelm85/master ...
1dce3ef on Aug 13
213 commits
docs update URLs for migration to jump-dev 4 months ago
notebook re-ran NQueens notebook 2 months ago
script pulled latest changes before pushing geo clustering changes 2 months ago
src review comments 6 months ago
test run with julia 10 2 months ago
.gitignore Update .gitignore 7 months ago
.travis.yml Update Travis CI builds 6 months ago
LICENSE Files generated by PkgTemplates 17 months ago
Manifest.toml run with julia 10 2 months ago
Project.toml pulled latest changes before pushing geo clustering changes 2 months ago
About
Tutorials on using JuMP for mathematical optimization in Julia
notebooks
mathematical-programming
julia
Readme
MIT License
Releases
No releases published
Packages
No packages published

The Julia Programming Language | JuliaPro – Julia Computing | Think Julia: How to Think Like a C | Introduction · JuMP | Julia Jump tutorial - gtsirogianni | GitHub - jump-dev/JuMPTutorial

github.com/jump-dev/JuMPTutorials.jl

Table of Contents