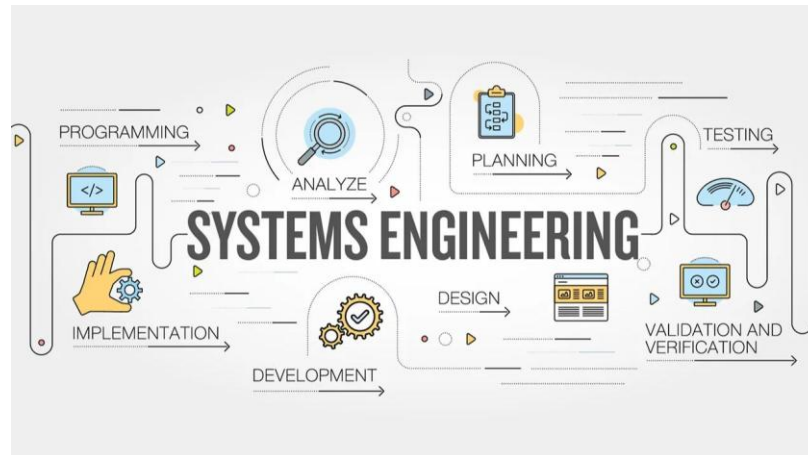


Διάλεξη 13

Ολοκληρωμένη Μοντελοποίηση και Σχεδίαση Πραγματικού Συστήματος

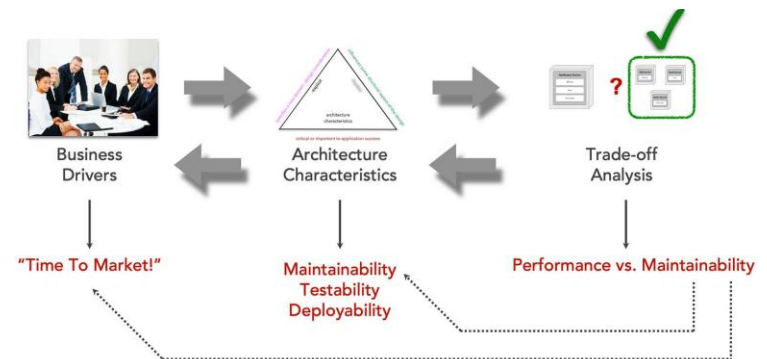
1. Από Μεμονωμένα Προβλήματα σε Ολοκληρωμένο Σύστημα

- Στις προηγούμενες διαλέξεις:
- – ορίσαμε μεταβλητές
- – κατασκευάσαμε περιορισμούς
- – αναπαραστήσαμε δίκτυα
- – κατασκευάσαμε και βελτιώσαμε λύσεις
- – συγκρίναμε πολλαπλά κριτήρια
- Σε αυτή τη διάλεξη:
- – συνδυάζουμε όλα τα προηγούμενα στοιχεία σε μία ολοκληρωμένη διαδικασία σχεδίασης



2. Η Πραγματική Σχεδίαση Συστημάτων

- Στην πράξη:
- – τα προβλήματα δεν εμφανίζονται χωρισμένα
- Ένα πραγματικό σύστημα περιέχει:
- – αποφάσεις
- – περιορισμούς
- – δίκτυα
- – πολλαπλά κριτήρια
- Η σχεδίαση απαιτεί:
- – συνδυασμό όλων αυτών των στοιχείων



3. Παράδειγμα: Σύστημα Διανομής Drone

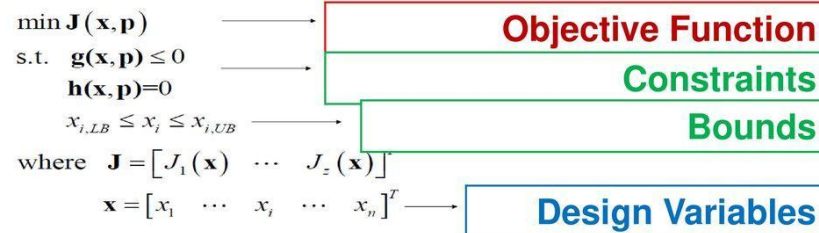
- Θέλουμε να σχεδιάσουμε:
- – σύστημα διανομής με drones
- Το σύστημα περιλαμβάνει:
- – αποθήκες
- – πελάτες
- – διαδρομές
- – περιορισμούς βάρους και χρόνου
- Το πρόβλημα:
- – συνδυάζει πολλά επιμέρους υποπροβλήματα



4. Βήμα 1 — Ορισμός Μεταβλητών

- Πρέπει να οριστούν:
 - οι αποφάσεις του συστήματος
- Παραδείγματα:
 - $x_{ij} = 1$ αν χρησιμοποιηθεί διαδρομή $i \rightarrow j$
 - $z_k = 1$ αν ενεργοποιηθεί αποθήκη k
 - b = χωρητικότητα μπαταρίας
- Οι μεταβλητές:
 - αναπαριστούν πραγματικές επιλογές σχεδίασης

OPTIMIZATION



5. Βήμα 2 — Ορισμός Περιορισμών

- Το σύστημα δεν μπορεί να λειτουργεί χωρίς περιορισμούς
- Παραδείγματα:
 - – περιορισμός ενέργειας
 - – περιορισμός βάρους
 - – περιορισμός χρόνου
 - – περιορισμός αριθμού drones
 - – περιορισμός προϋπολογισμού
- Οι περιορισμοί:
 - – καθορίζουν τον εφικτό χώρο λύσεων



6. Από το Μοντέλο στο Λογισμικό

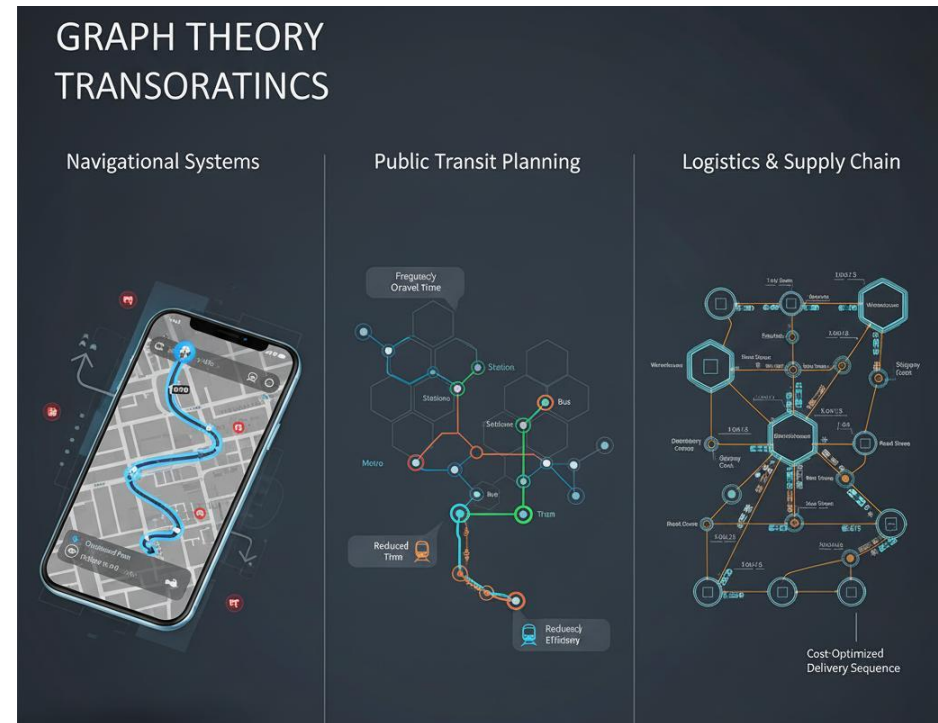
- Τα προβλήματα αυτά μπορούν να υλοποιηθούν σε:
- – Excel Solver
- – Gurobi
- – CPLEX
- Τα εργαλεία αυτά:
- – διαχειρίζονται μεταβλητές και περιορισμούς
- – ελέγχουν εφικτότητα
- – αναζητούν καλές λύσεις

	A	B	C	D	E	F
1	Variant	Number	Assembly	Polish	Pack	Profit
2	1	0	2	3	2	1.5
3	2	0	4	2	3	2.5
4	3	26250	3	3	2	3
5	4	0	7	4	5	4.5
6	Total		78750	78750	52500	78750
7	Sum		210000			
8	Limit		210000			
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						

Solver Results	
Solver found a solution. All constraints and optimality conditions are satisfied.	
☒ Keep Solver Solution	Reports
☐ Restore Original Values	Answer
	Sensitivity
	Limits
OK	Cancel
	Save Scenario...
	Help

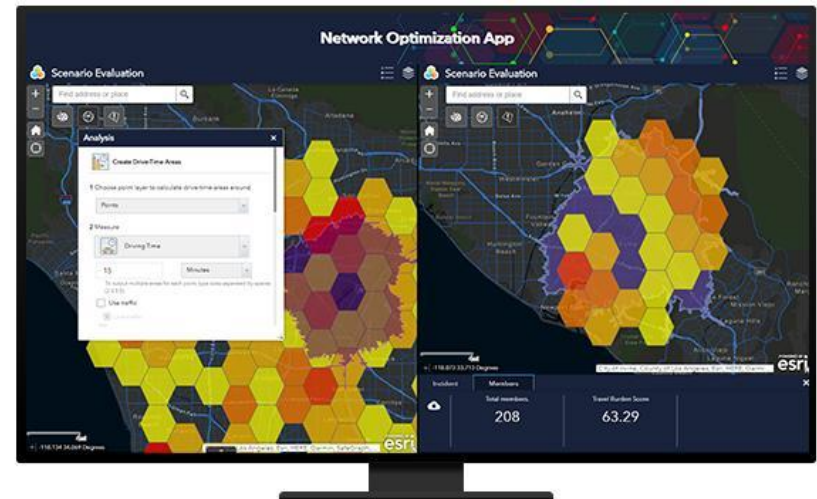
7. Βήμα 3 — Αναπαράσταση ως Δίκτυο

- Το πρόβλημα μπορεί να αναπαρασταθεί ως γράφημα
- Κόμβοι:
 - αποθήκες και πελάτες
- Ακμές:
 - πιθανές διαδρομές πτήσης
- Βάρη:
 - χρόνος
 - απόσταση
 - ενεργειακό κόστος



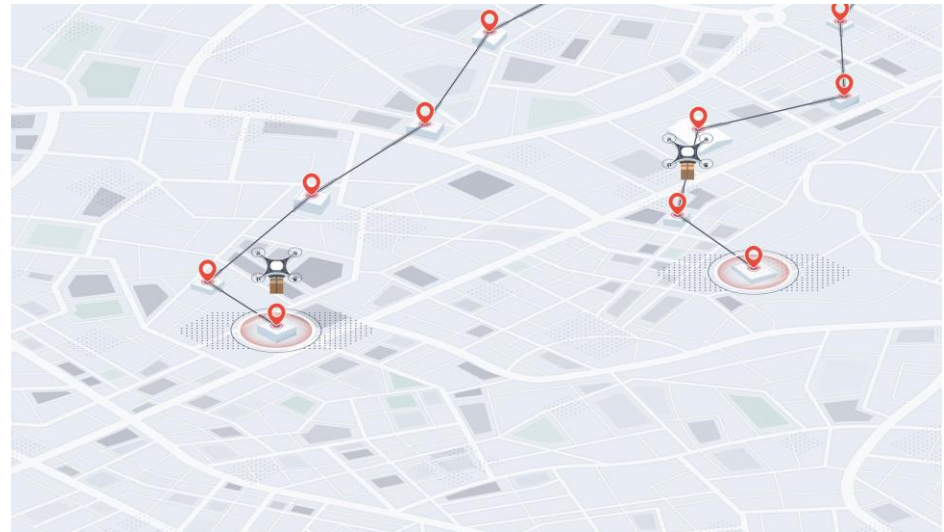
8. Υπολογιστική Αναπαράσταση Δικτύων

- Δίκτυα και διαδρομές μπορούν να αναπαρασταθούν υπολογιστικά
- Βιβλιοθήκες και εργαλεία:
 - – NetworkX (Python)
 - – ArcGIS Network Analyst
 - – Gephi
- Τα εργαλεία αυτά:
 - – υπολογίζουν διαδρομές
 - – αναλύουν συνδεσιμότητα
 - – αξιολογούν δομές δικτύων



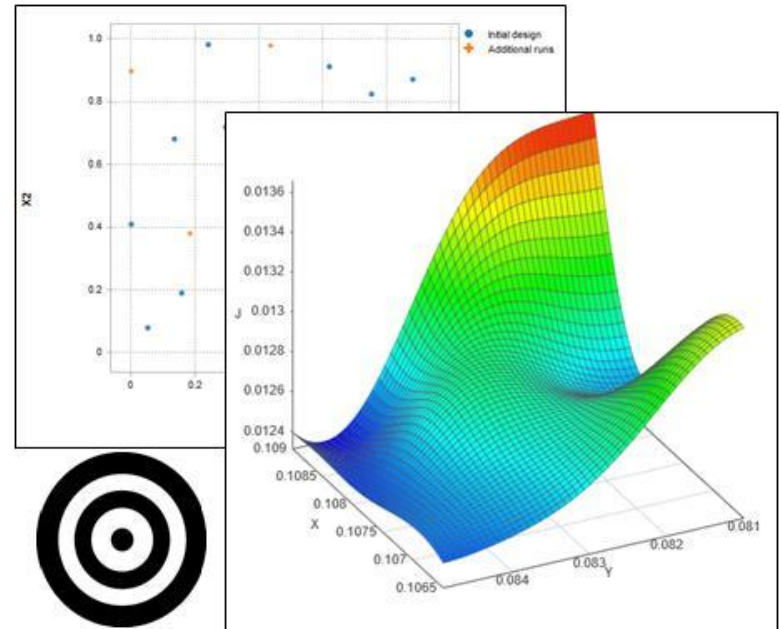
9. Βήμα 4 — Κατασκευή Αρχικής Λύσης

- Το λογισμικό πρέπει πρώτα να δημιουργήσει αρχική λύση
- Παράδειγμα:
 - αρχικές διαδρομές drones
 - αρχική ανάθεση παραδόσεων
- Η αρχική λύση:
 - δεν είναι απαραίτητα βέλτιστη
- Πρέπει όμως:
 - να είναι εφικτή



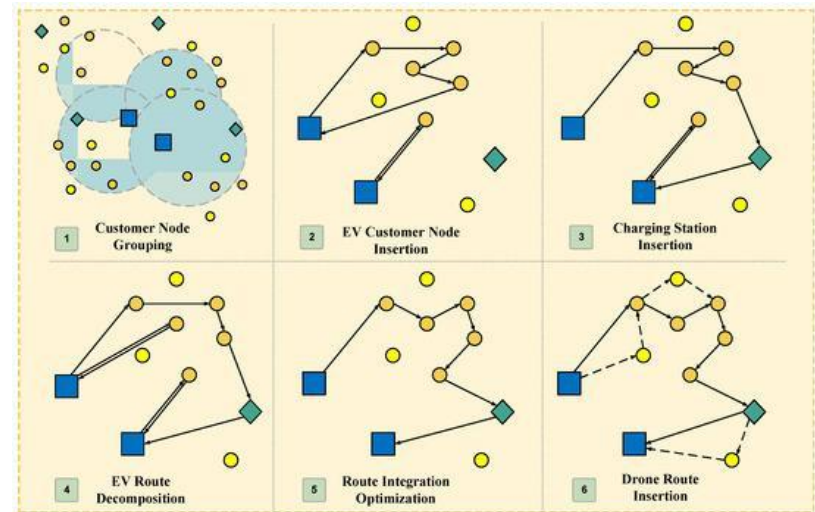
10. Βήμα 5 — Υπολογισμός Ποιότητας

- Για κάθε λύση υπολογίζεται η συνολική ποιότητα
- Παραδείγματα:
 - – συνολικό κόστος
 - – χρόνος παράδοσης
 - – κατανάλωση ενέργειας
 - – αξιοπιστία
- Η αξιολόγηση:
 - – επιτρέπει σύγκριση εναλλακτικών λύσεων



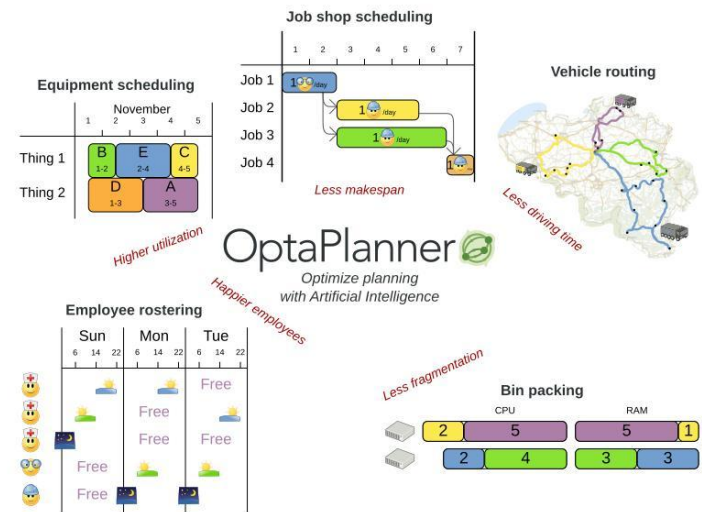
11. Βήμα 6 — Τοπική Βελτίωση

- Το πρόγραμμα τροποποιεί μικρά μέρη της λύσης
- Παραδείγματα:
 - – αλλαγή διαδρομής
 - – αλλαγή ανάθεσης drone
 - – αλλαγή αποθήκης εξυπηρέτησης
- Αν η νέα λύση:
 - – είναι καλύτερη και παραμένει εφικτή
- τότε γίνεται αποδεκτή



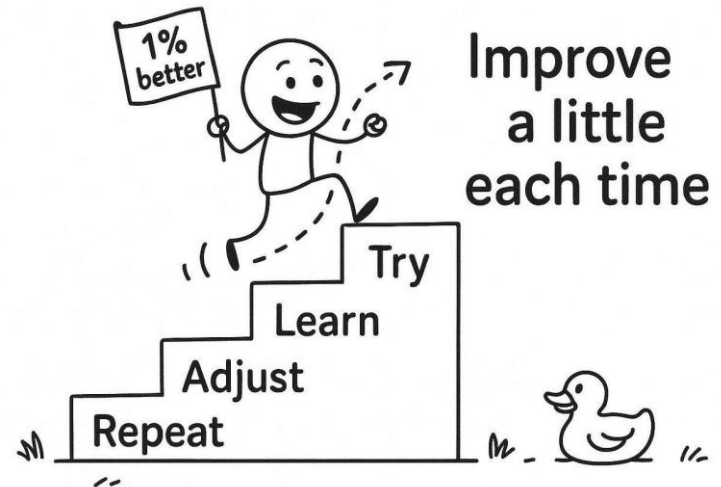
12. Υλοποίηση Τοπικής Αναζήτησης

- Σε μεγάλα προβλήματα χρησιμοποιούνται ευρετικές μέθοδοι
- Λογισμικά:
 - – Google OR-Tools
 - – OptaPlanner
 - – AnyLogic
- Τα συστήματα αυτά:
 - – δημιουργούν αρχικές λύσεις
 - – δοκιμάζουν μικρές αλλαγές
 - – συγκρίνουν αποτελέσματα επαναληπτικά



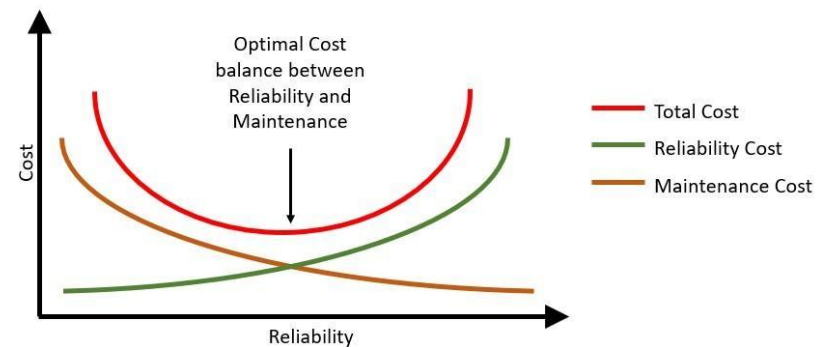
13. Η Λογική της Επαναληπτικής Αναζήτησης

- Η διαδικασία επαναλαμβάνεται πολλές φορές
- Σε κάθε επανάληψη:
 - δημιουργείται νέα υποψήφια λύση
- Το λογισμικό:
 - συγκρίνει συνεχώς αποτελέσματα
- Η λύση:
 - βελτιώνεται σταδιακά



14. Πολλαπλά Κριτήρια στη Σχεδίαση

- Το σύστημα δεν αξιολογείται μόνο με ένα κριτήριο
- Πιθανές προτεραιότητες:
 - χαμηλό κόστος
 - μικρός χρόνος
 - χαμηλή κατανάλωση
 - υψηλή αξιοπιστία
- Συνεπώς:
 - απαιτείται πολυκριτηριακή αξιολόγηση



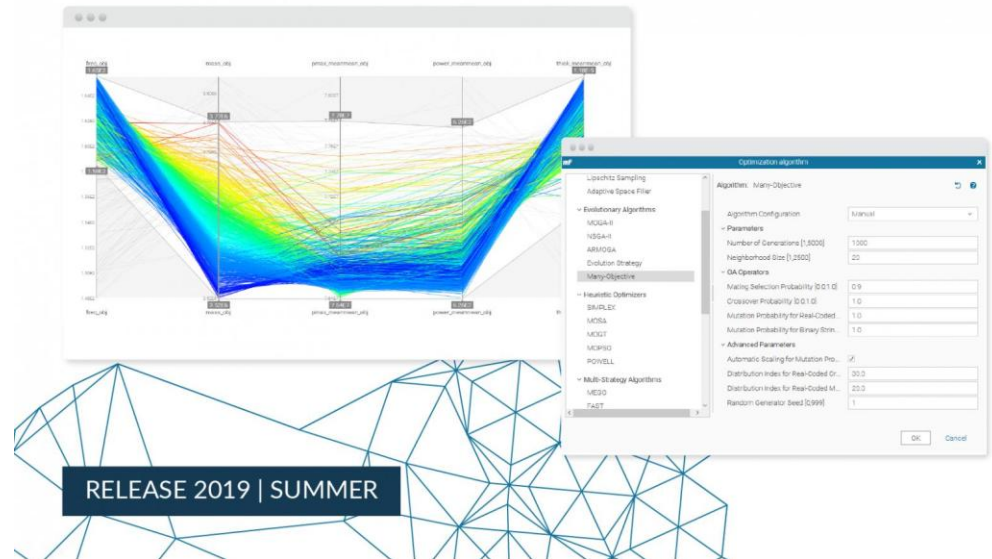
15. Υπολογιστική Πολυκριτηριακή Αξιολόγηση

- Το λογισμικό μπορεί να αξιολογεί πολλές διαστάσεις ταυτόχρονα
- Για κάθε λύση:
 - υπολογίζει διαφορετικά κριτήρια
- Στη συνέχεια:
 - συνδυάζει τα αποτελέσματα σε συνολικό σκορ
- Η διαδικασία:
 - οργανώνει τη λήψη αποφάσεων

	Criteria Weight	Alternative 1		Alternative 2		Alternative 3		Alternative 4		Alternative 5	
		Score	Weighted Score	Score	Weighted Score	Score	Weighted Score	Score	Weighted Score	Score	Weighted Score
Criterion 1	0.3	3	0.9	2	0.6	4	1.2	4	1.2	4	1.2
Criterion 2	0.25	2	0.5	3	0.75	3	0.75	4	1	2	0.5
Criterion 3	0.25	2	0.5	2	0.5	3	0.75	1	0.25	3	0.75
Criterion 4	0.15	3	0.45	4	0.6	3	0.45	2	0.3	4	0.6
Criterion 5	0.05	5	0.25	2	0.1	4	0.2	3	0.15	4	0.2
Total	1		2.6		2.55		3.35		2.9		3.25

16. Πολυκριτηριακά Εργαλεία

- Λογισμικά:
- – MATLAB
- – pymoo
- – modeFRONTIER
- Τα εργαλεία αυτά:
- – συγκρίνουν πολλές λύσεις
- – υπολογίζουν trade-offs
- – αναλύουν Pareto λύσεις



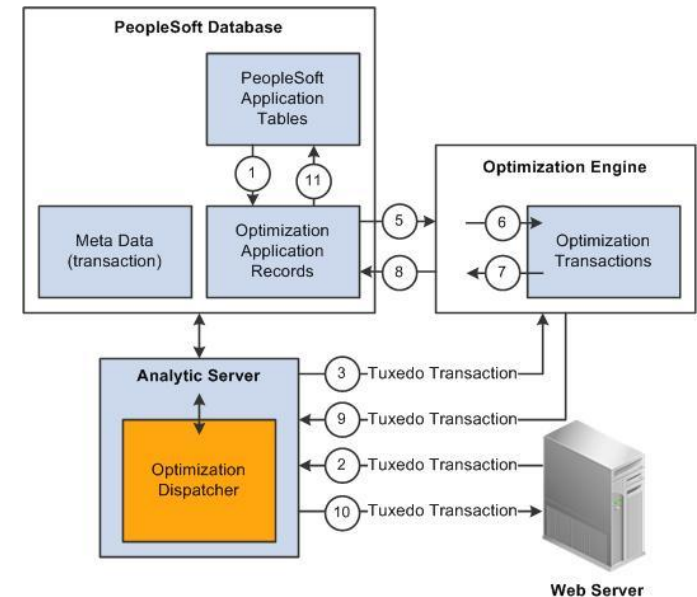
17. Παράδειγμα Πλήρους Υπολογιστικής Ροής

- Εισαγωγή δεδομένων
- Δημιουργία μαθηματικού μοντέλου
- Κατασκευή αρχικής λύσης
- Έλεγχος περιορισμών
- Υπολογισμός ποιότητας
- Τοπική βελτίωση
- Πολυκριτηριακή αξιολόγηση
- Επιλογή τελικής λύσης



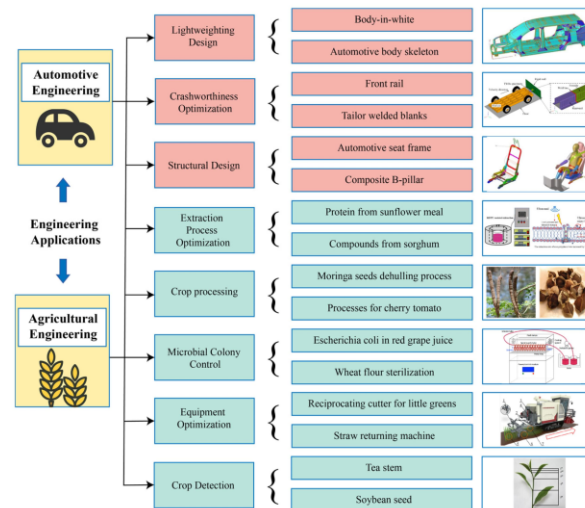
18. Τι Θα Περιείχε Ένα Πραγματικό Σύστημα

- Ένα πραγματικό λογισμικό θα περιείχε:
- – βάση δεδομένων
- – μοντέλο δικτύου
- – σύστημα αξιολόγησης
- – αλγόριθμο αναζήτησης
- – διαδικασία σύγκρισης λύσεων
- Η επίλυση:
- – γίνεται οργανωμένη υπολογιστική διαδικασία



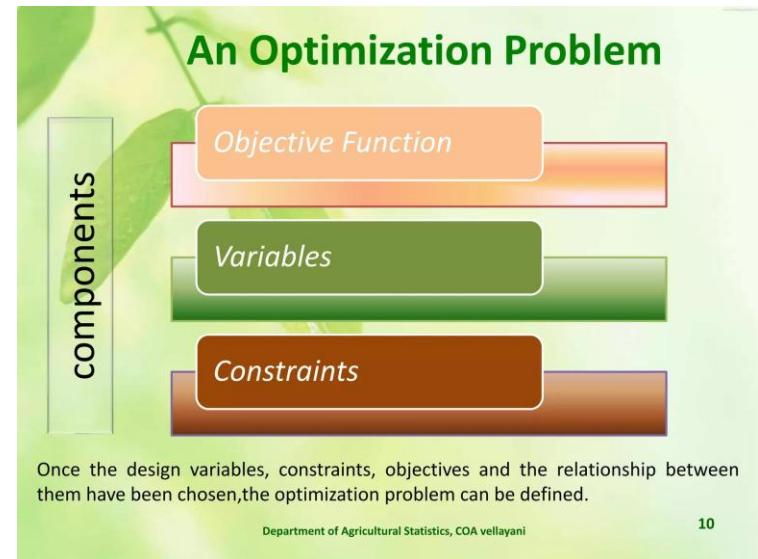
19. Από τη Μαθηματική Δομή στη Μηχανική Σχεδίαση

- Οι μαθηματικές μεταβλητές:
 - αντιστοιχούν σε πραγματικές αποφάσεις
- Οι περιορισμοί:
 - αντιστοιχούν σε φυσικά ή οικονομικά όρια
- Η βελτιστοποίηση:
 - γίνεται εργαλείο σχεδίασης πραγματικών συστημάτων



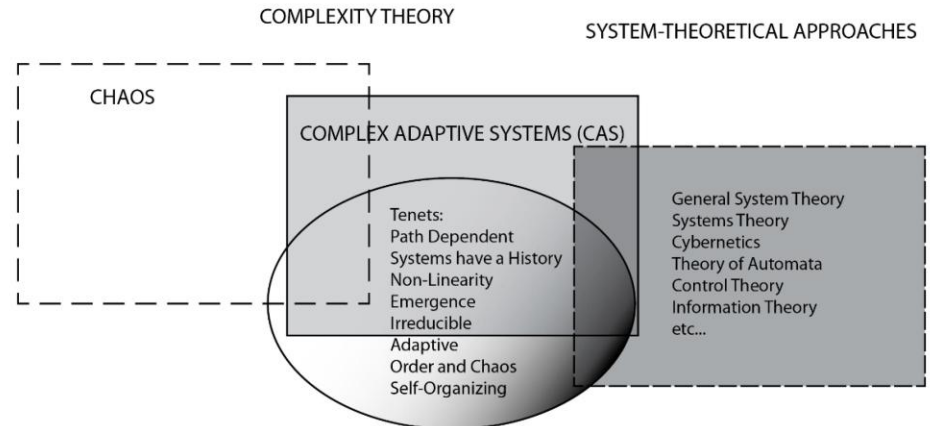
20. Σχεδίαση υπό Περιορισμούς

- Στην πράξη δεν υπάρχει απεριόριστη ελευθερία σχεδίασης
- Υπάρχουν:
 - οικονομικά όρια
 - τεχνολογικά όρια
 - ενεργειακά όρια
 - χρονικοί περιορισμοί
- Η σχεδίαση:
 - γίνεται διαδικασία επιλογών υπό περιορισμούς



21. Γιατί Χρησιμοποιούμε Βελτιστοποίηση

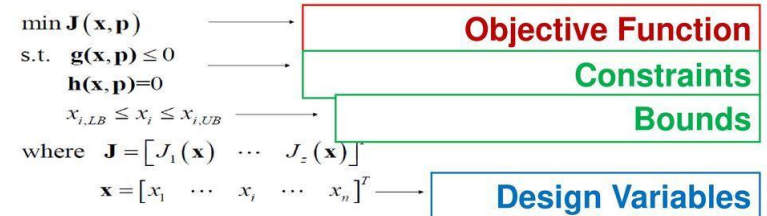
- Η πολυπλοκότητα των συστημάτων:
 - ξεπερνά συχνά τη διαισθητική ανάλυση
- Η βελτιστοποίηση:
 - οργανώνει τη διαδικασία λήψης αποφάσεων
- Επιτρέπει:
 - συστηματική αξιολόγηση
 - σύγκριση εναλλακτικών
 - τεκμηριωμένες αποφάσεις



22. Τι Έμαθε το Μάθημα

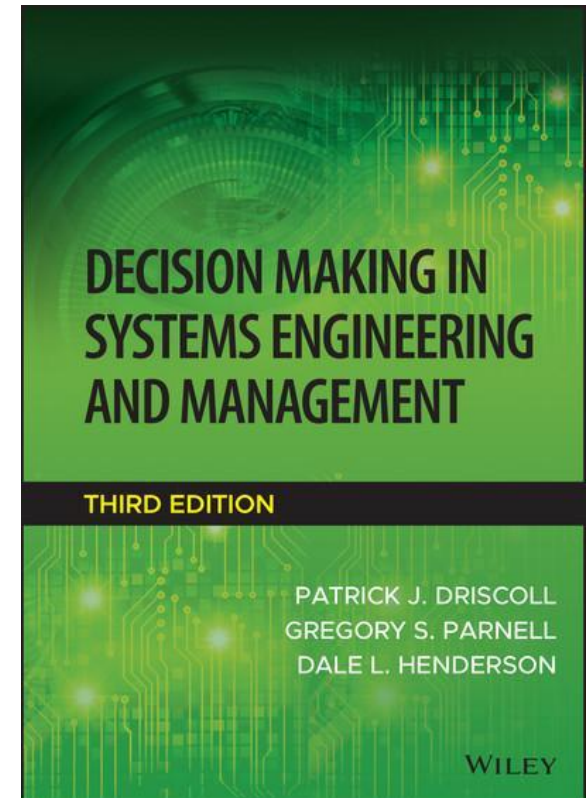
- Μετατροπή πραγματικών προβλημάτων:
- – σε μαθηματικά μοντέλα
- Ορισμός:
- – μεταβλητών
- – περιορισμών
- – αντικειμενικών συναρτήσεων
- Κατασκευή και βελτίωση λύσεων
- Πολυκριτηριακή αξιολόγηση αποφάσεων

OPTIMIZATION



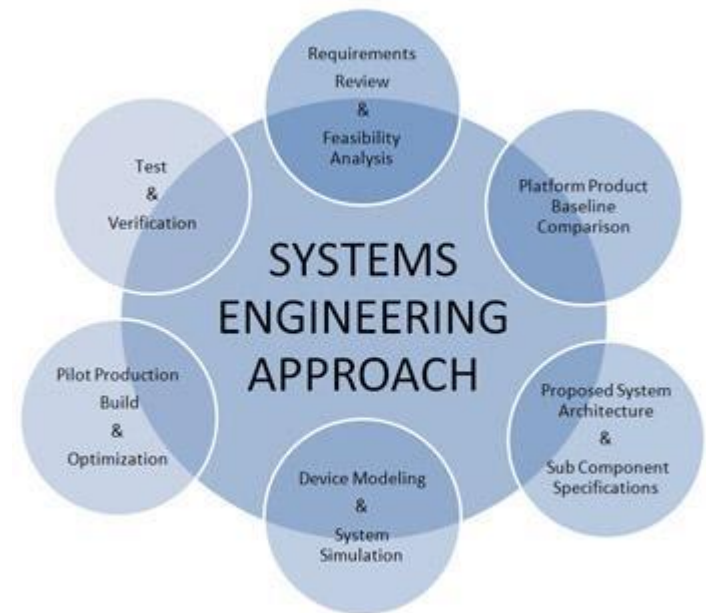
23. Από τη Διαίσθηση στη Συστηματική Απόφαση

- Η σχεδίαση δεν βασίζεται μόνο στη διαίσθηση
- Απαιτεί:
 - – δομή
 - – υπολογισμό
 - – αξιολόγηση
 - – σύγκριση
- Η βελτιστοποίηση:
 - – υποστηρίζει τεκμηριωμένη μηχανική απόφαση



24. Το Μάθημα ως Μεθοδολογία Σκέψης

- Το μάθημα δεν αφορά μόνο μαθηματικούς τύπους
- Αφορά:
 - τρόπο σκέψης
 - οργάνωση σύνθετων προβλημάτων
 - ανάλυση συστημάτων
 - λογική λήψης αποφάσεων
- Η βελτιστοποίηση:
 - αποτελεί εργαλείο μηχανικής σχεδίασης



25. Κεντρικό Μήνυμα Μαθήματος

- Κάθε πραγματικό σύστημα:
- – περιέχει αποφάσεις και περιορισμούς
- Η ποιότητα του σχεδιασμού:
- – εξαρτάται από τη δομή των αποφάσεων
- Η βελτιστοποίηση:
- – οργανώνει την αναζήτηση καλών λύσεων
- Η μηχανική σχεδίαση:
- – είναι διαδικασία συστηματικής λήψης αποφάσεων

