

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΒΙΟΜΗΧΑΝΙΑΣ



ΥΠΟΥΡΓΕΙΟ ΕΘΝΙΚΗΣ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΕΑΕΚ
ΕΥΡΩΠΑΪΚΗ ΕΝΩΣΗ
ΣΥΓΧΡΗΜΑΤΟΔΟΤΗΣΗ
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
ΕΥΡΩΠΑΪΚΟ ΤΑΜΕΙΟ ΠΕΡΙΦΕΡΕΙΑΚΗΣ ΑΝΑΠΤΥΞΗΣ



Η ΠΑΙΔΕΙΑ ΣΤΗΝ ΚΟΡΥΦΗ
Επιχειρησιακό Πρόγραμμα
Εκπαίδευσης και Αρχικής
Επαγγελματικής Κατάρτισης



ΑΝΑΜΟΡΦΩΣΗ ΠΡΟΠΤΥΧΙΑΚΟΥ
ΠΡΟΓΡΑΜΜΑΤΟΣ ΣΠΟΥΔΩΝ

ΣΥΝΟΠΤΙΚΗ ΕΚΘΕΣΗ

«ΑΝΑΠΤΥΞΗ ΚΑΙΝΟΤΟΜΩΝ ΕΡΓΑΛΕΙΩΝ ΓΙΑ ΤΗΝ ΕΠΙΛΥΣΗ
ΠΡΟΒΛΗΜΑΤΩΝ ΒΑΣΙΣΜΕΝΩΝ ΣΤΗ ΜΕΘΟΔΟΛΟΓΙΑ GOAL
BASED SCENARIO ΜΕ ΤΗ ΒΟΗΘΕΙΑ ΤΗΣ ΤΕΧΝΟΛΟΓΙΑΣ
CLIENT-SERVER»

Συντάκτης: ΜΑΣΤΡΟΓΙΑΝΝΙΔΟΥ ΧΡΙΣΤΙΝΑ

ΒΟΛΟΣ
ΜΑΡΤΙΟΣ 2007

ΠΕΡΙΕΧΟΜΕΝΑ

1. ΕΙΣΑΓΩΓΗ.....	3
2. ΓΡΑΜΜΙΚΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ.....	5
3. ΜΕΘΟΔΟΛΟΓΙΑ GOAL-BASED SCENARIO.....	8
4. ΔΙΑΤΥΠΩΣΗ ΠΡΟΒΛΗΜΑΤΟΣ.....	10
5. ΜΟΝΤΕΛΟ ΕΠΙΛΥΣΗΣ ΣΕ AMPL.....	12
6. ΔΙΕΠΙΦΑΝΕΙΑ ΧΡΗΣΗΣ (USER INTERFACE).....	15
7. ΣΥΜΠΕΡΑΣΜΑΤΑ.....	21

1. ΕΙΣΑΓΩΓΗ

Στα πλαίσια του έργου “Αναμόρφωση Προπτυχιακού Προγράμματος Σπουδών” του Τμήματος Μηχανολόγων Μηχανικών Βιομηχανίας του Πανεπιστημίου Θεσσαλίας προτείνουμε την Ανάπτυξη Καινοτόμων Εργαλείων για την Επίλυση Προβλημάτων Βασισμένων στη Μεθοδολογία Goal-Based Scenario με τη Βοήθεια της Τεχνολογίας Client Server. Το συγκεκριμένο υπόεργο αφορά την πιλοτική εφαρμογή της μεθόδου διδασκαλίας Goal-Based Scenario, η οποία και θα εφαρμοστεί για τους σκοπούς του μαθήματος του Γραμμικού Προγραμματισμού.

Πιο συγκεκριμένα, προτείνουμε την ανάπτυξη μίας διεπιφάνειας χρήσης (user interface), η οποία θα αποτελεί ένα φιλικό και διαδραστικό περιβάλλον που θα μπορεί ο κάθε φοιτητής να εισάγει τα δεδομένα εύκολα και γρήγορα. Στη συνέχεια, με τη βοήθεια ενός μοντέλου επίλυσης τέτοιων προβλημάτων σε γλώσσα προγραμματισμού AMPL, θα παρέχεται η βέλτιστη λύση του προβλήματος. Η διεπιφάνεια θα επικοινωνεί με την AMPL και θα παράγει τα αποτελέσματα, τα οποία θα διαβάζονται από την AMPL και θα επιστρέφονται στο χρήστη σε ευανάγνωστη μορφή. Με τον τρόπο αυτό, θα μπορεί ο κάθε ενδιαφερόμενος με απλό, εύχρηστο και γρήγορο τρόπο, όχι μόνο να επιλύει παρόμοια προβλήματα χωρίς να χρειάζεται να γνωρίζει κάποια γλώσσα προγραμματισμού, αλλά και να δημιουργεί σενάρια τα οποία θα έχει τη δυνατότητα να αξιολογήσει μέσω του λογισμικού και να δει κατά πόσο ανταποκρίνονται στην πραγματικότητα ή παρέχουν τη βέλτιστη λύση.

Το μοντέλο αυτό θα βασίζεται στην τεχνολογία client-server προσβάσιμη από το internet και θα είναι ανοιχτό για τους φοιτητές του Τμήματος Μηχανολόγων Μηχανικών Βιομηχανίας του Πανεπιστημίου Θεσσαλίας, αλλά και πιθανόν για τους φοιτητές τμημάτων όλης της Ελλάδας που ασχολούνται με το αντικείμενο αυτό.

Στην επόμενη παράγραφο, περιγράφεται το υπόβαθρο της μεθοδολογίας goal-based scenario και η χρησιμότητα του σε προβλήματα βελτιστοποίησης. Στην τρίτη παράγραφο παρουσιάζεται το πρόβλημα που μελετήσαμε, το οποίο αποτελεί ένα κλασσικό πρόβλημα σχεδιασμού παραγωγής (production planning). Στην τέταρτη παράγραφο, παρουσιάζεται το μοντέλο επίλυσης για το οποίο χρησιμοποιήθηκε η

γλώσσα μαθηματικού προγραμματισμού AMPL. Στην πέμπτη παράγραφο, παρουσιάζεται η διεπιφάνεια χρήσης και τα πλεονεκτήματα αυτής. Τέλος, στην έκτη παράγραφο παρουσιάζονται τα συμπεράσματα και οι επιπτώσεις που εκτιμάται να έχει στη διαδικασία διδασκαλίας σε προπτυχιακά τμήματα των Τριτοβάθμιων Εκπαιδευτικών Ιδρυμάτων.

2. ΓΡΑΜΜΙΚΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

Ο Γραμμικός προγραμματισμός είναι το δημοφιλέστερο μοντέλο στο χώρο της επιχειρησιακής έρευνας αλλά και της διοικητικής επιστήμης γενικότερα. Θα μπορούσαμε να τον περιγράψουμε ως μία τεχνική που ασχολείται με το πρόβλημα της κατανομής των περιορισμένων πόρων ή μέσων ενός συστήματος σε εναλλακτικές και ανταγωνιστικές μεταξύ τους δραστηριότητες κατά τον καλύτερο δυνατό τρόπο. Θεωρείται σαν μία από τις πιο σπουδαίες μαθηματικές ανακαλύψεις των μέσων χρόνων του εικοστού αιώνα και στις μέρες μας αποτελεί ένα μοντέλο ευρείας χρήσης για καθημερινά ζητήματα των περισσότερων μεσαίου και μεγάλου μεγέθους εμπορικών-βιομηχανικών εταιρειών¹.

Σε ένα πρόβλημα Γραμμικού Προγραμματισμού διακρίνουμε την αντικειμενική συνάρτηση και τους περιορισμούς. Τα προβλήματα Γραμμικού Προγραμματισμού είναι προβλήματα βελτιστοποίησης στα οποία τόσο η αντικειμενική συνάρτηση όσο και οι περιορισμοί είναι γραμμικές εξισώσεις. Η γενική διατύπωση ενός προβλήματος γραμμικού προγραμματισμού είναι η εξής:

$$\begin{aligned} \max z &= c_1 x_1 + c_2 x_2 + \dots + c_n x_n \\ \text{με τους περιορισμούς:} \\ a_{11} x_1 + a_{12} x_2 + \dots + a_{1n} x_n &= b_1 \\ a_{21} x_1 + a_{22} x_2 + \dots + a_{2n} x_n &= b_2 \\ &\cdot \\ &\cdot \\ &\cdot \\ a_{m1} x_1 + a_{m2} x_2 + \dots + a_{mn} x_n &= b_m \\ x_i \geq 0, b_j \geq 0 & (i = 1, 2, \dots, n \text{ και } j = 1, 2, \dots, m) \end{aligned}$$

Το πρόβλημα επίλυσης ενός συστήματος γραμμικών ανισώσεων χρονολογείται περίπου στην εποχή του Fourier. Ο Γραμμικός Προγραμματισμός ανακαλύφθηκε με τη μορφή μαθηματικού μοντέλου κατά τη διάρκεια του 2^{ου} Παγκοσμίου Πολέμου για να σχεδιαστούν τα έξοδα και οι επιστροφές με τέτοιο τρόπο ώστε να ελαχιστοποιηθεί

¹ Ν.Δ. Τσάντας, Π.Χ.Γ. Βασιλείου. Εισαγωγή στην Επιχειρησιακή Έρευνα, Εκδόσεις Ζήτη, Ιανουάριος 2000, Θεσσαλονίκη

το κόστος του στρατού και να αυξηθούν οι απώλειες του αντιπάλου. Μετά το 2^ο Παγκόσμιο Πόλεμο, πολλές βιομηχανίες κατάλαβαν τη χρησιμότητα του στον καθημερινό σχεδιασμό των διαδικασιών.

Η αρχική μαθηματική διατύπωση του προβλήματος καθώς και μία συστηματική διαδικασία επίλυσης του, η μέθοδος Simplex, οφείλεται στον G. Dantzig το 1947. Επίσης, σημαντικοί ιδρυτές του γραμμικού προγραμματισμού θεωρούνται ο John von Neumann ο οποίος ανέπτυξε τη θεωρία του δυισμού την ίδια χρονιά και ο Leonid Kantorovich, Ρώσος μαθηματικός, ο οποίος χρησιμοποιούσε παρόμοιες μεθόδους στα οικονομικά πριν τον Dantzig και κέρδισε το βραβείο Νόμπελ στα οικονομικά το 1975. Σημαντική ανακάλυψη μετά το Γραμμικό Προγραμματισμό θεωρείται η εισαγωγή της μεθόδου Interior Point Method, το 1984 από τους Fiacco και McCormick².

Η μέθοδος Simplex είναι μία επαναληπτική τεχνική για την επίλυση προβλημάτων γραμμικού προγραμματισμού και βασίζεται στη συστηματική δημιουργία βασικών εφικτών λύσεων και έλεγχο της αρίστευσής τους. Συγκεκριμένα ξεκινώντας από μία γνωστή βασική εφικτή λύση x^0 , δημιουργεί μία ακολουθία γειτονικών βασικών εφικτών λύσεων x^1, x^2, \dots, x^k με $c'x^0 \leq c'x^1 \leq \dots \leq c'x^k$, η οποία συγκλίνει στη βέλτιστη λύση του προβλήματος (αν υπάρχει βέβαια) σε πεπερασμένο αριθμό επαναλήψεων. Σημαντικό στοιχείο της μεθόδου αποτελεί επιπλέον και το γεγονός ότι εντοπίζει και τις περιπτώσεις στις οποίες το πρόβλημα είναι αδύνατο ή έχει λύση με μη πεπερασμένη τιμή της αντικειμενικής συνάρτησης.

Το πρωτότυπο παράδειγμα του Dantzig ο οποίος βρήκε τη βέλτιστη ανάθεση εβδομήντα ατόμων σε εβδομήντα εργασίες εξηγεί ακόμη την επιτυχία του. Η υπολογιστική δύναμη που χρειαζόταν για την επεξεργασία όλων των συνδυασμών ήταν τεράστια. Ο Dantzig παρατήρησε ότι η εύρεση της βέλτιστης λύσης έπαιρνε ελάχιστο χρόνο με τη μέθοδο Simplex, η οποία κατέστη η αφορμή για να παρατηρήσει ότι η λύση βρίσκεται πάντα στις κορυφές του πολυγώνου που σχηματίζουν οι εξισώσεις των περιορισμών.

² http://en.wikipedia.org/wiki/Linear_programming

Σήμερα, ο Γραμμικός Προγραμματισμός αποτελεί ένα σημαντικό πεδίο βελτιστοποίησης που χρησιμοποιείται από τους επιχειρησιακούς ερευνητές και τους αναλυτές προβλημάτων απόφασης. Πολλά πρακτικά προβλήματα της επιχειρησιακής έρευνας μπορούν να διατυπωθούν ως προβλήματα Γραμμικού Προγραμματισμού. Ειδικές περιπτώσεις προβλημάτων Γραμμικού Προγραμματισμού όπως τα προβλήματα ροής σε δίκτυα θεωρούνται αρκετά σημαντικά και έχουν αναπτυχθεί αρκετοί αλγόριθμοι επίλυσης τους. Ένας σημαντικός αριθμός αλγορίθμων προβλημάτων βελτιστοποίησης δουλεύει επιλύοντας σαν υποπροβλήματα γραμμικά προβλήματα. Επίσης, ο Γραμμικός Προγραμματισμός χρησιμοποιείται ευρέως στη διοίκηση επιχειρήσεων, είτε για να μεγιστοποιήσει το κέρδος είτε για να ελαχιστοποιήσει το κόστος παραγωγής. Μερικά παραδείγματα εφαρμογής του είναι στο συνδυασμό φαγητών (πρόβλημα της διαίτας), στη διαχείριση αποθεμάτων, στη διαχείριση χαρτοφυλακίου επενδύσεων, στην κατανομή των πόρων, στο σχεδιασμό διαφήμισης προϊόντων ευρείας κυκλοφορίας, κτλ.

3. ΜΕΘΟΔΟΛΟΓΙΑ GOAL-BASED SCENARIO

Η μεθοδολογία goal-based scenario, η οποία θα μπορούσε να μεταφραστεί στα ελληνικά ως “σενάρια βάσει στόχων”, προτάθηκε αρχικά από τον καθηγητή Roger Shank το 1992. Η μεθοδολογία βασίζεται στην ανάπτυξη σύνθετων καθοδηγούμενων προσομοιώσεων οι οποίες αποτελούν αναπαραστάσεις των υπό μελέτη διαδικασιών ή συστημάτων, με στόχο τον πειραματισμό και την εκτίμηση λειτουργικών τους χαρακτηριστικών. Η μεθοδολογία αυτή επιτρέπει στους φοιτητές να πειραματιστούν, να ανακαλύψουν και να εξηγήσουν επιστημονικές σχέσεις που διέπουν το πρόβλημα, να προβλέψουν γεγονότα και να αναπτύξουν προσόντα που θα τους φανούν χρήσιμα στη διαδικασία δοκιμάζω και αποτυγχάνω. Η βασική φιλοσοφία, είναι να εισαγάγει ένα δύσκολο αντικείμενο ξεκινώντας από τον τελικό στόχο και επιτρέποντας στον φοιτητή να πειραματισθεί.

Σήμερα, οι περισσότερες διδακτικές μέθοδοι συνδέονται άμεσα ή έμμεσα με τους Η/Υ και τις εποπτικές δυνατότητες που αυτοί προσφέρουν, με αποτέλεσμα να εκσυγχρονίζουν την αντίστοιχη διδακτική και εργαστηριακή διδασκαλία. Τα εργαλεία αυτά χρησιμοποιούνται σε συνδυασμό με κλασσικές μεθόδους διδασκαλίας για την καλύτερη παροχή γνώσης στους φοιτητές. Παρόλο που η χρήση του διαδικτύου (internet) προσφέρει την δυνατότητα για εύκολη πρόσβαση στην πληροφορία και την καλή παρουσίαση της, η πιο σημαντική συμβολή στην εκπαιδευτική διαδικασία σχετίζεται με τα κίνητρα που η συγκεκριμένη πληροφορία δίνει και το ενδιαφέρον που προκαλεί στους φοιτητές. Συγκεκριμένα περιβάλλοντα ανάπτυξης δίνουν την δυνατότητα δημιουργίας Goal Based Scenarios³. Πρόκειται για μια σειρά από σενάρια, το καθένα από τα οποία απαιτεί από τον φοιτητή την διεκπεραίωση μιας ξεκάθαρης και πλήρους αποστολής μέσω μιας σειράς ενεργειών, που είναι ειδικά σχεδιασμένες για την διδασκαλία και την καλλιέργεια συγκεκριμένων ικανοτήτων. Η διαδικασία αυτή παρέχει στους φοιτητές κίνητρα και ερεθίσματα για να ασχοληθούν και να συλλέξουν πληροφορίες και υλικό με τελικό σκοπό την ολοκλήρωση της συγκεκριμένης εργασίας, που τους έχει ανατεθεί.

³ Schank, R.C. and Kass, A. (1996): A Goal-Based Scenario for High-School Students, Communications of the ACM, vol. 39, No 4, 28-29.

Παράλληλα μέσα από το κατάλληλο υπολογιστικό περιβάλλον επισημαίνονται οι λανθασμένες επιλογές του φοιτητή και του δίνεται η δυνατότητα να επαναλάβει την διαδικασία με καλύτερες αρχικές επιλογές. Η αξιολόγηση των φοιτητών έχει σχέση με την ικανότητα τους να πετύχουν τους στόχους της αποστολής. Έτσι μέσα στην μεθοδολογία αυτή συμπεριλαμβάνονται η εκμάθηση, η εξάσκηση και η αξιολόγηση, πράγματα που στο παραδοσιακό σύστημα εκπαίδευσης απαρτίζουν τρεις διαφορετικές φάσεις.

Ένα παράδειγμα αυτού του είδους διδασκαλίας είναι το “δελτίο ειδήσεων”. Στα πλαίσια του μαθήματος κοινωνικών σπουδών στα λύκεια των ΗΠΑ, ο κάθε μαθητής πρέπει να δημιουργήσει το δικό του δελτίο ειδήσεων. Κατάλληλη εφαρμογή-πρόγραμμα του παρέχει σύντομες περιγραφές διαφόρων θεμάτων, υλικό σχετιζόμενο με το κάθε θέμα, μία λίστα ερωτημάτων με απαντήσεις από ειδικούς ή πρόσωπα, που σχετίζονται με τις υποθέσεις καθώς και πλήκτρα ελέγχου για τις ενέργειες σύνταξης του κειμένου. Με την ολοκλήρωση του δελτίου, ακολουθούν σχόλια και κριτικές από τους ειδικούς που διατυπώνουν τις απόψεις τους ανάλογα με την θέση τους και το τι εκπροσωπούν. Τελικός στόχος είναι να αναπτυχθεί η προσωπική άποψη του μαθητή έχοντας συμπεριλάβει όσο το δυνατό περισσότερα δεδομένα.

4. ΔΙΑΤΥΠΩΣΗ ΠΡΟΒΛΗΜΑΤΟΣ

Το πρόβλημα που μελετήσαμε, είναι ένα κλασσικό πρόβλημα σχεδιασμού παραγωγής (production planning). Σκοπός είναι η ελαχιστοποίηση του συνολικού κόστους μίας μονάδας παραγωγής χάλυβα οπλισμού σκυροδέματος για την οποία θα είναι γνωστές οι παραγγελίες για διάστημα έξι μηνών και θα συνδυάζονται οι έννοιες του κόστους παραγωγής του χάλυβα, αποθήκευσης, τα ημερομίσθια των εργατών καθώς και τα κόστη εκπαίδευσης και απόλυσης νέων εργατών που είναι πιθανό να προσληφθούν. Το πρόβλημα λοιπόν έχει ως εξής:

Η εταιρεία CONCRETE REINFORCING STEEL γνωρίζει ότι για τους επόμενους έξι μήνες οι παραγγελίες που θα έχει σε τόνους χάλυβα θα είναι:

Ιανουάριος	Φεβρουάριος	Μάρτιος	Απρίλιος	Μάιος	Ιούνιος
12000	10000	13000	5000	16000	18000

Τα κόστη παραγωγής του χάλυβα τους αντίστοιχους μήνες έχει εκτιμηθεί από την εταιρεία ότι μεταβάλλονται κάθε μήνα χωρίς όμως να επηρεάζονται από την παραγωγή του εργοστασίου τον κάθε μήνα και είναι τα ακόλουθα σε euro/τόνο χάλυβα:

PC1	PC2	PC3	PC4	PC5	PC6
5.05	4.85	4.73	4.46	5.24	5.60

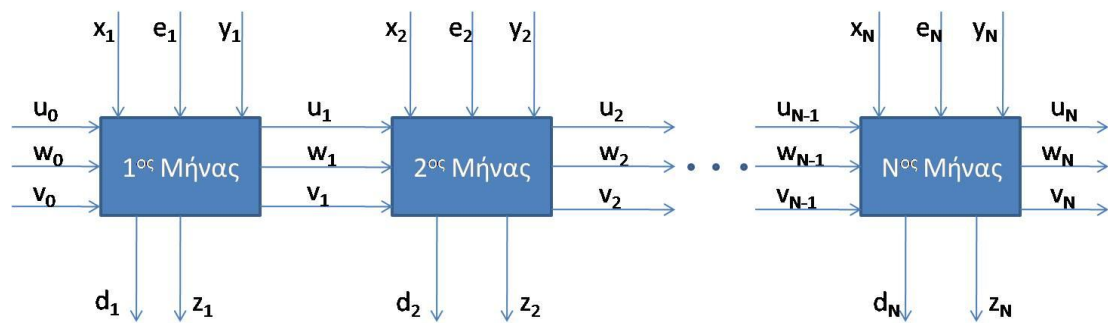
Το κόστος αγοράς της πρώτης ύλης (scrap) για την παραγωγή του χάλυβα οπλισμού σκυροδέματος την προβλεπόμενη περίοδο είναι σε euro/τόνο χάλυβα:

BC1	BC2	BC3	BC4	BC5	BC6
2.01	1.95	1.98	1.95	1.87	1.89

Η εταιρεία επίσης υπολογίζει ότι για την παραγωγή 1000 τόνων χάλυβα ανά μήνα απαιτούνται 12 εργάτες, οι οποίοι εργάζονται τις προβλεπόμενες ώρες χωρίς να υπάρχει η ανάγκη υπερωριών. Ακόμη ο μισθός κάθε εργάτη είναι 850 euro ανά μήνα εργασίας. Στην περίπτωση που το εργοστάσιο αποφασίσει να προσλάβει επιπλέον εργατικό δυναμικό τότε σημειώνεται πως το κόστος εκπαίδευσης κάθε νέου εργάτη είναι 1050 euro ενώ το κόστος απόλυσης του είναι 1750 euro. Επίσης το κόστος αποθήκευσης για διάστημα ενός μήνα ανά τόνο πρώτης ύλης είναι 0.05 euro ενώ αντίστοιχα το κόστος αποθήκευσης έτοιμου προϊόντος είναι 0.97 euro/τόνο. Τέλος δίνεται στον φοιτητή η πληροφορία ότι το εργοστάσιο ξεκινάει την 1^η Ιανουαρίου με

120 εργάτες δυναμικό, 20000 τόνους πρώτης ύλης και 3000 τόνους έτοιμου προς παράδοση προϊόντος στις αποθήκες τους, ενώ την 1^η Ιουλίου θα πρέπει να έχει στις αποθήκες του 8000 τόνους πρώτης ύλης και 5000 τόνους έτοιμου προϊόντος, και ζητείται από αυτόν να καθορίσει την πολιτική του εργοστασίου αναφέροντας τους εργάτες που θα εργαστούν τον κάθε μήνα, την ποσότητα πρώτης ύλης που θα αγοράσει και την ποσότητα που θα παράγει κάθε χρονική περίοδο ώστε να ελαχιστοποιήσει το συνολικό κόστος για τους έξι αυτούς μήνες.

Το πρόβλημα αυτό θα μπορούσε να εκφραστεί ως ένα ισοζύγιο των διαφόρων παραμέτρων που υπεισέρχονται στο πρόβλημα. Η παρακάτω εικόνα δείχνει τη ροή των διαφόρων παραμέτρων.



Όπου:

x_i : η ποσότητα παραγωγής το μήνα i

u_i : η ποσότητα αποθήκευσης έτοιμου προϊόντος το μήνα i

v_i : η ποσότητα αποθήκευσης πρώτης ύλης το μήνα i

y_i : η ποσότητα αγοράς πρώτης ύλης το μήνα i

w_i : ο αριθμός των εργατών το μήνα i

e_i : ο αριθμός των εργατών που προσλαμβάνονται το μήνα i

z_i : ο αριθμός των εργατών που απολύονται το μήνα i

5. ΜΟΝΤΕΛΟ ΕΠΙΛΥΣΗΣ ΣΕ AMPL

Ο προγραμματισμός μεγάλων προβλημάτων μαθηματικού προγραμματισμού δεν περιλαμβάνει απλώς την ελαχιστοποίηση ή τη μεγιστοποίηση της αντικειμενικής συνάρτησης η οποία υπόκειται σε κάποιους περιορισμούς. Πριν εφαρμοστεί κάποιος αλγόριθμος για την επίλυσή τους, πρέπει να καταβάλουμε σημαντική προσπάθεια για τη διατύπωση του μοντέλου και τη δημιουργία της απαραίτητης δομής των δεδομένων.

Τα στάδια που πρέπει να γίνουν είναι διαδοχικά τα εξής:

- Η διατύπωση του μοντέλου (το σύνολο των μεταβλητών, η αντικειμενική συνάρτηση και οι περιορισμοί που εκφράζουν τη γενική μορφή του προβλήματος που θέλουμε να λύσουμε).
- Η συγκέντρωση δεδομένων που προσδιορίζουν μία ή περισσότερες ειδικές περιπτώσεις του προβλήματος.
- Η δημιουργία αντικειμενικής συνάρτησης και περιορισμών από το μοντέλο και τα δεδομένα.
- Η επίλυση του προβλήματος (εφαρμογή συγκεκριμένου αλγορίθμου ο οποίος βρίσκει τις βέλτιστες λύσεις για τις μεταβλητές).
- Η ανάλυση των αποτελεσμάτων.
- Η βελτίωση του μοντέλου και των δεδομένων και η επανάληψη της διαδικασίας επίλυσης.

Αν οι άνθρωποι μπορούσαν να διαχειριστούν τα προγράμματα με τον ίδιο τρόπο που τα διαχειρίζονται οι αλγόριθμοι, η διατύπωση του μοντέλου και η δημιουργία της αντικειμενικής συνάρτησης και των περιορισμών θα ήταν εύκολη και άμεση. Στην πραγματικότητα, υπάρχουν πολλές διαφορές ανάμεσα στον τρόπο που αντιλαμβάνονται το πρόβλημα οι προγραμματιστές και σε αυτόν που το λύνουν οι αλγόριθμοι. Η μετατροπή από τη μία μορφή στην άλλη είναι μία διαδικασία χρονοβόρα, και συχνά επιρρεπής σε λάθη.

Πολλές από τις δυσκολίες μετατροπής από τη μορφή του προγραμματιστή στη μορφή του αλγορίθμου μπορούν να ξεπεραστούν με τη χρήση κάποιας γλώσσας

μαθηματικού προγραμματισμού. Οι γλώσσες προγραμματισμού χρησιμοποιούνται για να εκφράσουν τη μορφή του προγραμματιστή με τέτοιο τρόπο ώστε να εισάγεται απευθείας στον υπολογιστή. Η μετατροπή μετέπειτα στη μορφή του αλγορίθμου, πρέπει να γίνεται εξ ολοκλήρου από τον υπολογιστή, χωρίς το ενδιάμεσο στάδιο του προγραμματισμού.

Συγκρινόμενη με προγενέστερες γλώσσες, η AMPL διακρίνεται για τη γενικότητα της σύνταξης και για τις ομοιότητες που παρουσιάζει η διατύπωση των εκφράσεων με την αλγεβρική διατύπωση. Παρέχει μία ποικιλία τύπων και λειτουργιών για τον καθορισμό των δεικτών των συνόλων, καθώς επίσης, και λογικών εκφράσεων. Η AMPL έχει βασιστεί πολύ στην πρωτοπόρα γλώσσα προγραμματισμού XML, αλλά ενσωματώνει πολλές διαφορές και επεκτάσεις.

Η διατύπωση του μοντέλου ξεκινά με τη δήλωση των συνόλων και των παραμέτρων. Στη συνέχεια, δηλώνονται οι μεταβλητές και τέλος ορίζονται η αντικειμενική συνάρτηση και οι περιορισμοί συναρτήσεων των συνόλων, των παραμέτρων και των μεταβλητών.

Η διατύπωση ενός μοντέλου δεν αποτελεί τη βελτιστοποίηση κάποιου συγκεκριμένου προβλήματος. Εάν θέλουμε να ορίσουμε κάποιο συγκεκριμένο πρόβλημα πρέπει να εισάγουμε στο μοντέλο συγκεκριμένες τιμές για τα σύνολα και τις παραμέτρους.

Η διαδικασία που ακολουθεί η AMPL κατά την επίλυση είναι η ανάγνωση του μοντέλου και των δεδομένων, η ανάλυσή τους και η επεξεργασία τους. Για να μπορέσει όμως, η AMPL να καταλάβει το μοντέλο και τα δεδομένα πρέπει αυτά να έχουν συγκεκριμένη μορφή. Παραδείγματος χάριν, η δήλωση των παραμέτρων πρέπει να αρχίζει με τη λέξη `param` και να τελειώνει με “;”. Μαθηματικές εκφράσεις όπως $a_{ki}x_{ji}$, $i \in R$ και $\sum_{t=1}^T$ αντικαθίστανται από εκφράσεις που χρησιμοποιούν αποκλειστικά χαρακτήρες ASCII.

Η AMPL επιτρέπει τη χρησιμοποίηση ονομάτων με πολλούς χαρακτήρες σε αντίθεση με τις αλγεβρικές εκφράσεις που χρησιμοποιούν συνήθως ένα μόνο

χαρακτήρα για τις μεταβλητές. Τα πέντε κυριότερα μέρη ενός αλγεβρικού μοντέλου, τα σύνολα, οι παράμετροι, οι μεταβλητές, η αντικειμενική συνάρτηση και οι περιορισμοί, αποτελούν και τα κυριότερα μέρη ενός μοντέλου στην AMPL.

Παρακάτω φαίνεται το μοντέλο στην AMPL το οποίο επιλύει το πρόβλημα που παρουσιάστηκε στην προηγούμενη παράγραφο.

```

param T>=0 integer;      # time period

param d{0..T};          #Demand
param pc{1..T};         #Production Cost
param sc{1..T};         #Scrap Cost
param p;                #worker payment
param tw;               #Worker training cost
param r;                #worker dismissal cost
param h;                #raw material inventory cost
param s;                #final product inventory cost
param ergates;
param tonoi;
param Ustart;
param Vstart;
param Wstart;
param Ufinal;
param Vfinal;

var X{0..T} integer >=0; #production on month i
var Y{0..T} integer >=0; #tons of raw material to purchase on
month i
var V{0..T} integer >=0; #tons of raw material stored every month
var U{0..T} integer >=0; #tons of final product stored every month
var W{0..T} integer >=0; #number of workers each month
var E{0..T} integer >=0; #number of workers been hired every month
var Z{0..T} integer >=0; #number of workers been dismissed every
month

minimize objective: sum {t in 1..T} pc[t]*X[t] + sum {t in 1..T}
sc[t]*Y[t] + sum {t in 1..T} h*V[t] + sum {t in 1..T} s*U[t] + sum
{t in 1..T} p*W[t] + sum {t in 1..T} tw*E[t] + sum {t in 1..T}
r*Z[t];

subject to workers1 {t in 1..T-1}: W[t]>=(ergates*X[t])/tonoi;

subject to rminvent {t in 1..T}: V[t] = V[t-1]+Y[t]-X[t];
subject to fpinvent {t in 1..T}: U[t] = U[t-1]+X[t]-d[t];
subject to workers2 {t in 1..T}: W[t] = W[t-1]+E[t]-Z[t];

subject to constr1: U[0]=Ustart;
subject to constr2: V[0]=Vstart;
subject to constr3: W[0]=Wstart;
subject to constr4: U[T]>=Ufinal;
subject to constr5: V[T]>=Vfinal;

subject to constr6: Y[0]=0;
subject to constr7: X[0]=0;

subject to constr8: sum {t in 1..T} X[t]+U[0] >= sum {t in 1..T}
d[t]+U[T];

```

6. ΔΙΕΠΙΦΑΝΕΙΑ ΧΡΗΣΗΣ (USER INTERFACE)

Ένα από τα πιο σημαντικά τμήματα κάθε εφαρμογής και ειδικότερα αυτών που βασίζονται στην τεχνολογία client-server και προορίζονται για το ευρύ κοινό είναι η Διεπιφάνεια Χρήσης ή User Interface (UI). Δηλαδή, το τμήμα εκείνο της εφαρμογής με το οποίο ο χρήστης έρχεται σε επαφή φυσικά και εννοιολογικά. Στην ουσία, η διεπιφάνεια χρήσης είναι μία γλώσσα εισόδου για τον χρήστη, μία γλώσσα εξόδου για τον υπολογιστή και ένα πρωτόκολλο για την αλληλεπίδρασή τους. Έτσι, ανεξάρτητα από την ποιότητα της εφαρμογής, ένα καλό UI είναι συχνά αυτό που θα ωθήσει τον χρήστη να ασχοληθεί με την εφαρμογή, ενώ αντίθετα ένα κακό UI θα τον αποθαρρύνει να την χρησιμοποιήσει.

Ένα καλό UI εκτός από το γεγονός ότι θα πρέπει να είναι εύχρηστο, χαρακτηρίζεται και από τα παρακάτω:

- να είναι εύκολο να μάθει κανείς να χειρίζεται το σύστημα
- η χρήση του συστήματος να είναι αποδοτική, δηλαδή το UI θα πρέπει να αναδεικνύει τις λειτουργίες του συστήματος
- να είναι εύκολο να θυμάται κανείς πώς να το χρησιμοποιεί
- να έχει χαμηλό ρυθμό σφαλμάτων. Καταστροφικά σφάλματα δεν πρέπει να υπάρχουν
- να είναι ευχάριστο στη χρήση

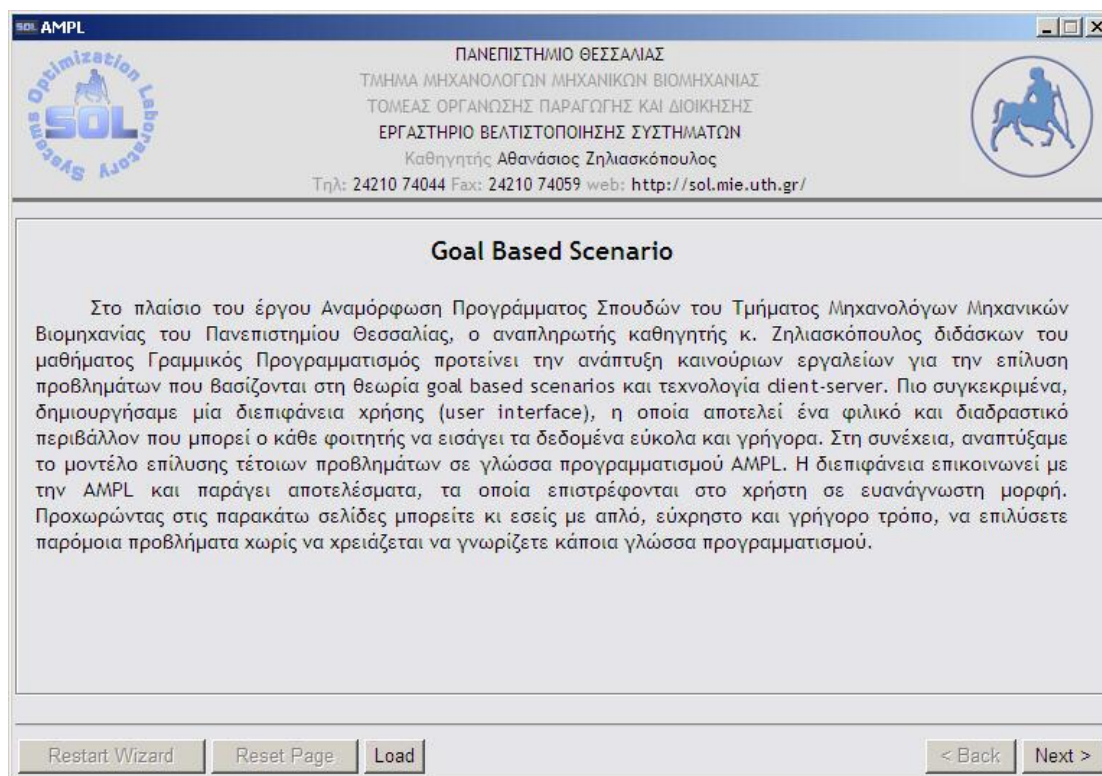
Ένα άλλο στοιχείο που λαμβάνεται σοβαρά υπ' όψη στα UIs είναι η εμφάνιση. Το πώς φαίνεται ένα UI είναι πολύ σημαντικό, καθώς η σωστή σχεδίασή του βοηθάει τον χρήστη να επικεντρώνει την προσοχή του κάθε φορά εκεί που χρειάζεται. Αυτό, ελαττώνει σημαντικά τον φόρτο εργασίας του χρήστη. Ακόμη, τα διάφορα μέρη στα οποία χωρίζεται το UI πρέπει να είναι ξεκάθαρα, όχι μόνο οπτικά αλλά και με βάση την λειτουργικότητά τους.

Το UI είναι, όπως είπαμε, το μέσο επικοινωνίας χρήστη και υπολογιστή. Έτσι, αναπόφευκτα, οι σχεδιαστές του θα πρέπει να λάβουν υπ' όψη τους το κοινό που θα χρησιμοποιήσει την εφαρμογή. Συγκεκριμένα, οι οδηγίες χρήσης που εμφανίζονται στην οθόνη πρέπει να είναι επαρκείς για να βοηθήσουν τον χρήστη να λύσει κάποιο πρόβλημα και να είναι στο κατάλληλο επίπεδο για το κοινό στο οποίο απευθύνεται η

εφαρμογή. Επίσης, τα μηνύματα λάθους πρέπει να είναι κατατοπιστικά και φιλικά, έτσι ώστε να μην «τρομοκρατούν» το χρήστη αλλά να τον βοηθάνε να εντοπίσει την πηγή του προβλήματος. Κάτι άλλο που πρέπει, επίσης, να επισημανθεί είναι η συμπεριφορά του UI κατά τη διάρκεια μεγάλων καθυστερήσεων του συστήματος. Η σωστή και ενδεδειγμένη συμπεριφορά είναι η γνωστοποίηση με κάποιο τρόπο στο χρήστη ότι το σύστημα εκτελεί κάποια χρονοβόρα λειτουργία. Μάλιστα, ακόμη καλύτερο θα ήταν αν η πρόοδος αυτής της λειτουργίας γινόταν εμφανής στον χρήστη.

Ένα άλλο χαρακτηριστικό απαραίτητο για την ευχρηστία και την φιλικότητα του UI είναι να μπορεί ο χρήστης να σώσει την πρόοδο της εργασίας του, έτσι ώστε την επόμενη φορά να συνεχίσει από το σημείο που σταμάτησε, χωρίς να χρειαστεί να ξανακάνει τα ίδια πράγματα. Μάλιστα, πολλά UI κάνουν αυτόματα αυτή τη δουλειά σε τακτά χρονικά διαστήματα για να προφυλάξουν το χρήστη από απώλεια μεγάλου μέρους της δουλειάς του σε περίπτωση που κάτι απρόσμενο συμβεί (π.χ. διακοπή ρεύματος ή λάθος χειρισμός από πλευράς χρήστη).

Στις παρακάτω εικόνες φαίνεται η διεπιφάνεια χρήσης που σχεδιάσαμε για το συγκεκριμένο πρόβλημα. Στην πρώτη εικόνα φαίνεται η εισαγωγή του προγράμματος.



Στην επόμενη εικόνα φαίνονται κάποιες γενικές πληροφορίες για το γραμμικό προγραμματισμό, την ιστορία και τις εφαρμογές του.

AMPL

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
 ΤΜΗΜΑ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΒΙΟΜΗΧΑΝΙΑΣ
 ΤΟΜΕΑΣ ΟΡΓΑΝΩΣΗΣ ΠΑΡΑΓΩΓΗΣ ΚΑΙ ΔΙΟΙΚΗΣΗΣ
 ΕΡΓΑΣΤΗΡΙΟ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ ΣΥΣΤΗΜΑΤΩΝ
 Καθηγητής Αθανάσιος Ζηλιασκόπουλος
 Τηλ: 24210 74044 Fax: 24210 74059 web: http://sol.mie.uth.gr/

Τι είναι ο Γραμμικός Προγραμματισμός

Ο Γραμμικός προγραμματισμός είναι το δημοφιλέστερο μοντέλο στο χώρο της επιχειρησιακής έρευνας αλλά και της διοικητικής επιστήμης γενικότερα. Θα μπορούσαμε να τον περιγράψουμε ως μία τεχνική που ασχολείται με το πρόβλημα της κατανομής των περιορισμένων πόρων ή μέσων ενός συστήματος σε εναλλακτικές και ανταγωνιστικές μεταξύ τους δραστηριότητες κατά τον καλύτερο δυνατό τρόπο. Θεωρείται σαν μία από τις πιο σπουδαίες μαθηματικές ανακαλύψεις των μέσων χρόνων του εικοστού αιώνα και στις μέρες μας αποτελεί ένα μοντέλο ευρείας χρήσης για καθημερινά ζητήματα των περισσότερων μεσαίου και μεγάλου μεγέθους εμπορικών-βιομηχανικών εταιρειών.

Σε ένα πρόβλημα Γραμμικού Προγραμματισμού διακρίνουμε την αντικειμενική συνάρτηση και τους περιορισμούς. Τα προβλήματα Γραμμικού Προγραμματισμού είναι προβλήματα βελτιστοποίησης στα οποία τόσο η αντικειμενική συνάρτηση όσο και οι περιορισμοί είναι γραμμικές εξισώσεις. Η γενική διατύπωση ενός προβλήματος γραμμικού προγραμματισμού είναι η εξής:

$$\text{Maximize } z = c_1x_1 + c_2x_2 + \dots + c_nx_n$$

κάτω από τις συνθήκες:

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1$$

Restart Wizard Reset Page < Back Next >

Στην παρακάτω εικόνα φαίνεται η διατύπωση του προβλήματος το οποίο θα προσπαθήσει ο χρήστης να επιλύσει με τη βοήθεια του συγκεκριμένου προγράμματος.

AMPL

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
 ΤΜΗΜΑ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΒΙΟΜΗΧΑΝΙΑΣ
 ΤΟΜΕΑΣ ΟΡΓΑΝΩΣΗΣ ΠΑΡΑΓΩΓΗΣ ΚΑΙ ΔΙΟΙΚΗΣΗΣ
 ΕΡΓΑΣΤΗΡΙΟ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ ΣΥΣΤΗΜΑΤΩΝ
 Καθηγητής Αθανάσιος Ζηλιασκόπουλος
 Τηλ: 24210 74044 Fax: 24210 74059 web: http://sol.mie.uth.gr/

Διατύπωση του Προβλήματος

Η εταιρεία CONCRETE REINFORCING STEEL γνωρίζει ότι για τους επόμενους έξι μήνες οι παραγγελίες που θα έχει σε τόνους κάλυβα θα είναι:

Ιανουάριος	Φεβρουάριος	Μάρτιος	Απρίλιος	Μάιος	Ιούνιος
12000	10000	13000	5000	16000	18000

Τα κόστη παραγωγής του κάλυβα τους αντίστοιχους μήνες έχει εκτιμηθεί από την εταιρεία ότι μεταβάλλονται κάθε μήνα χωρίς όμως να επηρεάζονται από την παραγωγή του εργοστασίου τον κάθε μήνα και είναι τα ακόλουθα σε ευρο/τόνο κάλυβα:

PC1	PC2	PC3	PC4	PC5	PC6
5.05	4.85	4.73	4.46	5.24	5.60

Το κόστος αγοράς της πρώτης ύλης (scrap) για την παραγωγή του κάλυβα οπλισμού σκυροδέματος την προβλεπόμενη περίοδο είναι σε ευρο/τόνο κάλυβα:

BC1	BC2	BC3	BC4	BC5	BC6

Restart Wizard Reset Page < Back Next >

Στην παρακάτω εικόνα φαίνεται η μορφοποίηση του προβλήματος που παρουσιάστηκε στην προηγούμενη διαφάνεια, όπου ξεχωρίζουμε την αντικειμενική συνάρτηση και τους περιορισμούς.

AMPL

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΤΜΗΜΑ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΒΙΟΜΗΧΑΝΙΑΣ
ΤΟΜΕΑΣ ΟΡΓΑΝΩΣΗΣ ΠΑΡΑΓΩΓΗΣ ΚΑΙ ΔΙΟΙΚΗΣΗΣ
ΕΡΓΑΣΤΗΡΙΟ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ ΣΥΣΤΗΜΑΤΩΝ
Καθηγητής Αθανάσιος Ζηλιασκόπουλος
Τηλ: 24210 74044 Fax: 24210 74059 web: http://sol.mie.uth.gr/

Μορφοποίηση του Προβλήματος

Αντικειμενική Συνάρτηση

Min [(5.05X₁ + 4.85X₂ + 4.73X₃ + 4.46X₄ + 5.24X₅ + 5.60X₆) + 0.97 (U₁ + U₂ + U₃ + U₄ + U₅ + U₆) + 0.05 (V₁ + V₂ + V₃ + V₄ + V₅ + V₆) + (2.01Y₁ + 1.95Y₂ + 1.98Y₃ + 1.95Y₄ + 1.87Y₅ + 1.89Y₆) + 850 (W₁ + W₂ + W₃ + W₄ + W₅ + W₆) + 1050 (E₁ + E₂ + E₃ + E₄ + E₅ + E₆) + 1750 (Z₁ + Z₂ + Z₃ + Z₄ + Z₅ + Z₆)]

Περιορισμοί

W_i ≥ (12 * X_i) / 1000 (1)

U_{i+1} = U_i + X_i - d_i (2)

V_{i+1} = V_i + Y_i - X_i (3)

W_{i+1} = W_i + E_{i+1} - Z_{i+1} (4)

Restart Wizard Reset Page < Back Next >

Στην επόμενη εικόνα παρατηρούμε την πιο κρίσιμη διαφάνεια του παρόντος προγράμματος, στην οποία ο χρήστης αλληλεπιδρά με το περιβάλλον προσπαθώντας να μαντέψει τη βέλτιστη λύση. Μόλις ο χρήστης συμπληρώσει τις τιμές που νομίζει ότι βελτιστοποιούν το πρόγραμμα παραγωγής της συγκεκριμένης βιομηχανίας, μπορεί να ελέγξει αν αυτή είναι όντως η βέλτιστη λύση (πατώντας το κουμπί «Έλεγχος») και αν όχι ποιοι περιορισμοί παραβιάζονται. Κάθε φορά που ο χρήστης ελέγχει την ορθότητα της λύσης του, ένας μετρητής δίπλα στο κουμπί «Έλεγχος» καταγράφει τον αριθμό των προσπαθειών του χρήστη.

Επίσης, ο χρήστης μπορεί να δει τη βέλτιστη λύση του συγκεκριμένου προβλήματος πατώντας το κουμπί «Λύση». Με το πάτημα του συγκεκριμένου κουμπιού, η διεπιφάνεια χρήσης δίνει εντολή στη γλώσσα μαθηματικού προγραμματισμού AMPL να παράγει τη βέλτιστη λύση και να την επιστρέφει στη διεπιφάνεια χρήσης. Κατά όμοιο τρόπο, δίπλα στο κουμπί «Λύση» υπάρχει ένας μετρητής, ο οποίος καταγράφει εάν και πόσες φορές είδε ο χρήστης τη βέλτιστη λύση.

AMPL Optimization Laboratory

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
 ΤΜΗΜΑ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΒΙΟΜΗΧΑΝΙΑΣ
 ΤΟΜΕΑΣ ΟΡΓΑΝΩΣΗΣ ΠΑΡΑΓΩΓΗΣ ΚΑΙ ΔΙΟΙΚΗΣΗΣ
 ΕΡΓΑΣΤΗΡΙΟ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ ΣΥΣΤΗΜΑΤΩΝ
 Καθηγητής Αθανάσιος Ζηλιασκόπουλος
 Τηλ: 24210 74044 Fax: 24210 74059 web: http://sol.mie.uth.gr/

Βήμα 2

Παρακαλώ εισάγετε στα παρακάτω κελιά τη λύση που νομίζετε ότι είναι η βέλτιστη. Αφού συμπληρώσετε όλα τα κελιά πατήστε Έλεγχος για δοκιμή ή Λύση για την εμφάνιση της βέλτιστης λύσης.

Τιμή Αντικειμενικής συνάρτησης: Έλεγχος (0) Λύση (0)

Μήνας	Ζήτηση	Παραγωγή	Αγορά Πρώτης Ύλης	Αποθήκη Πρώτης Ύλης	Αποθήκη Τελικού Προϊόντος	Αριθμός Εργατών	Αύξηση Αριθμού Εργατών	Μείωση Αριθμού Εργατών
0								
1	12000							
2	10000							
3	13000							
4	5000							
5	16000							
6	18000							

Restart Wizard Reset Page Save < Back Next >

Στην παρακάτω εικόνα φαίνεται πως είναι η συγκεκριμένη διαφάνεια αφού συμπληρώσουμε όλα τα κενά πεδία.

AMPL Optimization Laboratory

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
 ΤΜΗΜΑ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΒΙΟΜΗΧΑΝΙΑΣ
 ΤΟΜΕΑΣ ΟΡΓΑΝΩΣΗΣ ΠΑΡΑΓΩΓΗΣ ΚΑΙ ΔΙΟΙΚΗΣΗΣ
 ΕΡΓΑΣΤΗΡΙΟ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ ΣΥΣΤΗΜΑΤΩΝ
 Καθηγητής Αθανάσιος Ζηλιασκόπουλος
 Τηλ: 24210 74044 Fax: 24210 74059 web: http://sol.mie.uth.gr/

Βήμα 2

Παρακαλώ εισάγετε στα παρακάτω κελιά τη λύση που νομίζετε ότι είναι η βέλτιστη. Αφού συμπληρώσετε όλα τα κελιά πατήστε Έλεγχος για δοκιμή ή Λύση για την εμφάνιση της βέλτιστης λύσης.

Τιμή Αντικειμενικής συνάρτησης: 1354265 Έλεγχος (0) Λύση (1)

Μήνας	Ζήτηση	Παραγωγή	Αγορά Πρώτης Ύλης	Αποθήκη Πρώτης Ύλης	Αποθήκη Τελικού Προϊόντος	Αριθμός Εργατών	Αύξηση Αριθμού Εργατών	Μείωση Αριθμού Εργατών
0		0	0	20000	3000	120	0	0
1	12000	10000	0	10000	1000	120	0	0
2	10000	13000	3000	0	4000	156	36	0
3	13000	13250	13250	0	4250	159	3	0
4	5000	13250	13250	0	12500	159	0	0
5	16000	13250	13250	0	9750	159	0	0
6	18000	13250	21250	8000	5000	159	0	0

Restart Wizard Reset Page Save < Back Next >

Στη συνέχεια, πατώντας το κουμπί «Έλεγχος» βλέπουμε αυτό που μας επιστρέφει το πρόγραμμα. Πιο συγκεκριμένα, το πρόγραμμα μας ειδοποιεί εάν η λύση που πληκτρολογήσαμε είναι η βέλτιστη και αν όχι, μας ενημερώνει για τους περιορισμούς

που παραβιάζονται. Μας δίνει επίσης τη δυνατότητα να ξαναδοκιμάσουμε πατώντας το κουμπί «OK» ή να σταματήσουμε τις προσπάθειες πατώντας το κουμπί «Cancel».

The screenshot shows the AMPL software interface. At the top, it displays the logo of the Optimization Laboratory (SOL) and the name of the institution: ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ, ΤΜΗΜΑ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΒΙΟΜΗΧΑΝΙΑΣ, ΤΟΜΕΑΣ ΟΡΓΑΝΩΣΗΣ ΠΑΡΑΓΩΓΗΣ ΚΑΙ ΔΙΟΙΚΗΣΗΣ, ΕΡΓΑΣΤΗΡΙΟ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ ΣΥΣΤΗΜΑΤΩΝ, Καθηγητής Αθανάσιος Ζηλιασκόπουλος. Contact information is provided: Τηλ: 24210 74044 Fax: 24210 74059 web: http://sol.mie.uth.gr/.

The main window is titled "Βήμα 2" and contains a table for input data. A dialog box is overlaid on the table, displaying error messages:

AMPL
 Η λύση αυτή δεν είναι εφικτή ή δεν είναι βέλπστη
 *Δεν επαρκεί η παραγωγή για την ζήτηση 75000+3000 >= 74000+5000
 *Μήνας 6: Δεν επαρκεί το τελικό προϊόν για τη ζήτηση 5000=8750+13250-18000
 Αν θέλετε να ξαναδοκιμάσετε, πατήστε OK. Διαφορετικά πατήστε Cancel.

The table below shows the input data for months 0 to 6:

Μήνας	Ζήτηση	Παραγωγή	13250	1000	3250	159	Αύξηση Αριθμού Εργατών	Μείωση Αριθμού Εργατών
0		0					0	0
1	12000	9000					0	0
2	10000	13000					36	0
3	13000	13250	13250	1000	3250	159	3	0
4	5000	13250	13250	1000	11500	159	0	0
5	16000	13250	13250	1000	8750	159	0	0
6	18000	13250	21250	8000	5000	159	0	0

At the bottom of the window, there are buttons for "Restart Wizard", "Reset Page", "Save", "< Back", and "Next >".

Τέλος, σε περίπτωση που δεν θέλουμε να ξαναδοκιμάσουμε και πατήσουμε «Cancel», το πρόγραμμα θα μας οδηγήσει στη διαφάνεια που φαίνεται στην παρακάτω εικόνα, η οποία παρέχει στο χρήστη κάποιες χρήσιμες ιστοσελίδες όπου μπορεί να βρει περισσότερες πληροφορίες για το γραμμικό προγραμματισμό.

The screenshot shows the AMPL software interface. At the top, it displays the logo of the Optimization Laboratory (SOL) and the name of the institution: ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ, ΤΜΗΜΑ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΒΙΟΜΗΧΑΝΙΑΣ, ΤΟΜΕΑΣ ΟΡΓΑΝΩΣΗΣ ΠΑΡΑΓΩΓΗΣ ΚΑΙ ΔΙΟΙΚΗΣΗΣ, ΕΡΓΑΣΤΗΡΙΟ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ ΣΥΣΤΗΜΑΤΩΝ, Καθηγητής Αθανάσιος Ζηλιασκόπουλος. Contact information is provided: Τηλ: 24210 74044 Fax: 24210 74059 web: http://sol.mie.uth.gr/.

The main window contains the following text:

Αν θέλετε να μάθετε περισσότερες πληροφορίες για το Γραμμικό Προγραμματισμό και τις εφαρμογές του, επισκεφτείτε τις παρακάτω ιστοσελίδες.

- [Linear programming - Wikipedia, the free encyclopedia](#)
- [Optimization Technology Center](#)
- [Linear Programming FAQ](#)
- [GLPK \(GNU Linear Programming Kit\)](#)
- [Mathematics Archives - Topics in Mathematics - Linear/Nonlinear Programming](#)

Αν θέλετε να κλείσετε την εφαρμογή, πατήστε [εδώ](#).

At the bottom of the window, there are buttons for "Restart Wizard", "Reset Page", "< Back", and "Next >".

7. ΣΥΜΠΕΡΑΣΜΑΤΑ

Στη συγκεκριμένη δουλειά παρουσιάσαμε την ανάπτυξη καινοτόμων εργαλείων για την επίλυση προβλημάτων, βασισμένων στη μεθοδολογία goal-based scenario με τη βοήθεια της τεχνολογίας client-server. Πιο συγκεκριμένα, αναπτύξαμε μία εφαρμογή η οποία στοχεύει στη βελτιστοποίηση του προγραμματισμού παραγωγής μίας βιομηχανίας παραγωγής χάλυβα οπλισμού σκυροδέματος.

Η συγκεκριμένη εφαρμογή μπορεί εύκολα να εισάγει το χρήστη στην έννοια του γραμμικού προγραμματισμού και να του δίνει τη δυνατότητα να προσπαθήσει να επιλύσει ένα τέτοιο πρόβλημα χωρίς να είναι απαραίτητο να γνωρίζει τη θεωρία του γραμμικού προγραμματισμού ή μεθόδους επίλυσης. Προσπαθώντας ο χρήστης να «μαντέψει» τη βέλτιστη λύση μέσα από μία διαδικασία σωστού-λάθους, αποκτά ταυτόχρονα την εμπειρία και αντιλαμβάνεται το βαθύτερο νόημα και τη σημασία του γραμμικού προγραμματισμού και της χρησιμότητας του στην επίλυση παρόμοιων προβλημάτων. Βασικός στόχος της προσπάθειας αυτής στο πλαίσιο του έργου «Αναμόρφωση του Προπτυχιακού Προγράμματος Σπουδών», ήταν η ανάπτυξη των εργαλείων αυτών να εισάγει καινούριες μεθόδους διδασκαλίας, όπου με τη χρήση ηλεκτρονικών υπολογιστών να μπορούν οι φοιτητές να πειραματίζονται επάνω στην επίλυση τέτοιων προβλημάτων, ώστε να προκαλέσουμε το ενδιαφέρον τους και να παρακινήσουμε την ενεργή συμμετοχή τους στο μάθημα.