Pianificazione e Gestione dei Servizi Sanitari

Utilizzo di CPLEX

Prof. Domenico Conforti – Lez.20 - 14/11/2023 - Autori: Gianluca Pisani, Chiara Zanella
Revisionatori: Zanella – Pisani

I modelli che si vedranno in questa lezione sono i seguenti:

- L' **IBM ILOG CPLEX Optimization Studio**
- **IBM ILOG Script**

IBM ILOG CPLEX Optimization Studio: *è un modello per massimizzare l'efficienza e l'efficacia nei problemi decisionali. Si usano per i problemi che non hanno vincoli lineari, ma non lineari, con problemi decisionali più complessi. Il linguaggio di programmazione OPL si propone di fare ciò, attinge da informazioni essenziali per mezzo di numerose librerie di codici.*
Si può individuare una soluzione che non è ottima ma che è funzionale alla richiesta effettuata, con i modelli precedenti visti fino ad ora: con quello attuale, si raggiunge il massimo grado di ottimizzazione possibile.

Il Pacchetto (che il prof ha mostrato) è composto da:

- L'ottimizzatore matematico **CPLEX**
- Il linguaggio di programmazione dichiarativo **OPL**, con costrutti che richiamano l'algebra dei modelli di ottimizzazione
- Il linguaggio di programmazione procedurale **IBM ILOG Script**, utile per modificare il comportamento del risolutore. È una porzione di codice, assorbita da Java, che implementa opzioni di risoluzione di problemi
- Un **IDE**, ambiente di programmazione che supporta il programmatore nello sviluppo del codice

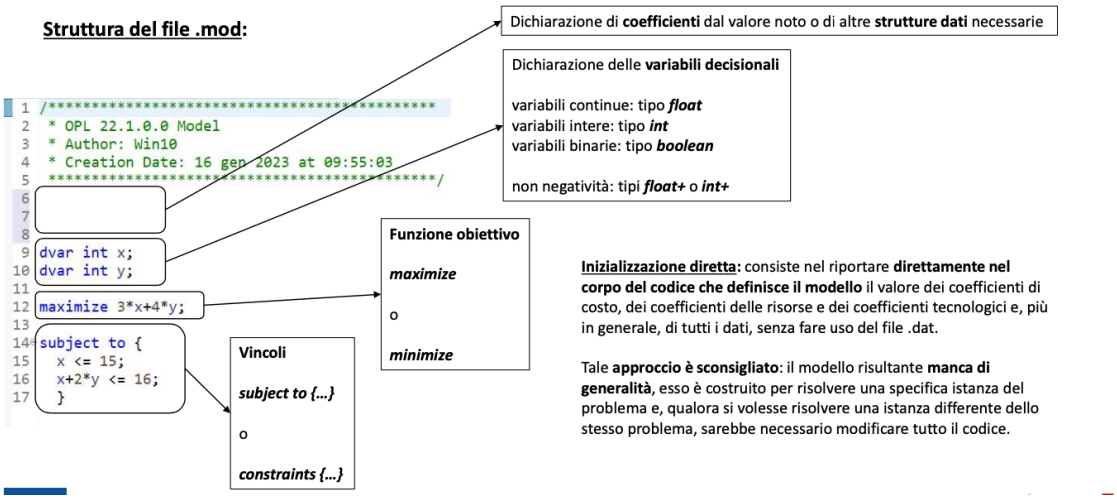
<u>Come si crea un nuovo progetto OPL?:</u>
File -> New -> OPL Project -> necessario a questo punto impostare:

- **Project Name** (obbligatorio, il nome deve essere univoco)
- Possibile introdurre una descrizione del progetto (facoltativa) in Description
- Già da qui, se si vuole, è possibile generare automaticamente altre parti del progetto utili nel seguito, spuntando: **Create Model**: crea il file dove dovrà essere scritto il modello
- **Create Data:** crea il file dove dovranno essere inseriti i dati della specifica istanza del problema
- **Create Settings:** crea il file dal quale è possibile impostare i parametri del risolutore
- **Add Run Configuration:** genera il file che permette di avviare il risolutore su una istanza del problema

Un primo semplice modello di ottimizzazione:

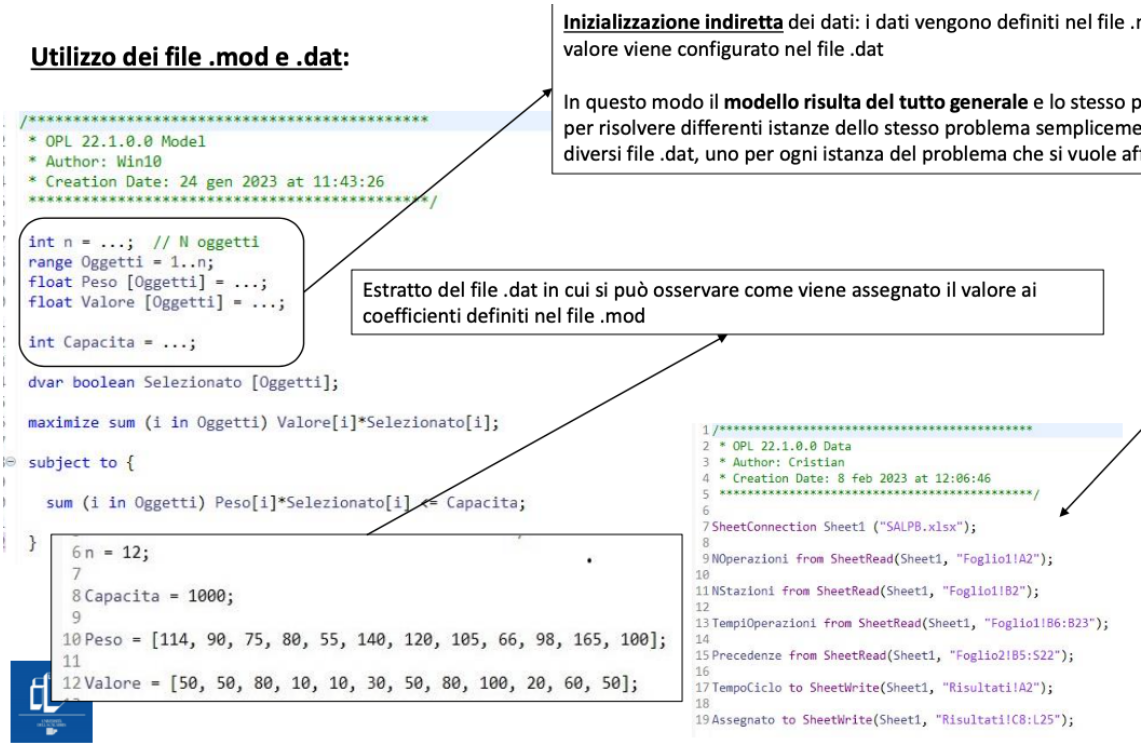
$$\min 3x + 4y$$
$$\text{s.t.}$$
$$x \geq 15$$
$$x + 2y \geq 16$$
$$x \geq 0, \text{ intero}$$
$$y \geq 0, \text{ intero}$$

STRUTTURA DEI FILE .MOD



In questo caso si utilizza l'**INIZIALIZZAZIONE DIRETTA** che consiste nel riportare direttamente nel corpo del codice che definisce il modello il valore dei coefficienti di costo, dei coefficienti delle risorse e dei coefficienti tecnologici e, più in generale, di tutti i dati, senza fare uso del file.dat. Tale approccio è sconsigliato: il modello risultante manca di generalità, esso è costruito per risolvere una specifica istanza del problema e, qualora si volesse risolvere una istanza differente dello stesso problema, sarebbe necessario modificare tutto il codice.

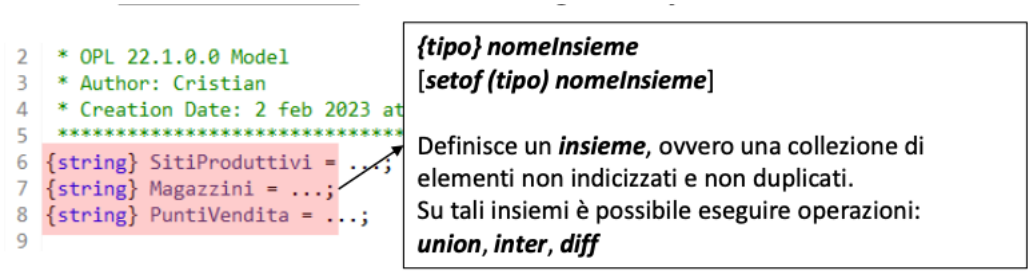
UTILIZZO FILE .MOD E .DAT



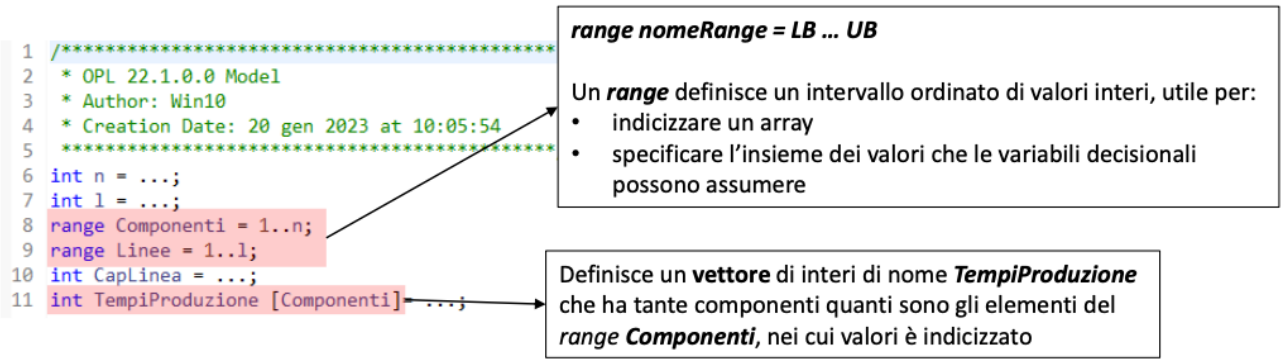
In questo caso si utilizza l'**INIZIALIZZAZIONE INDIRETTA** dei dati: i dati vengono definiti nel file .mod ma i loro valore viene configurato nel file.dat. In questo modo il modello risulta del tutto generale e lo stesso potrà essere utilizzato per risolvere differenti istanze dello stesso problema semplicemente generando diversi file .dat, uno per ogni istanza del problema che si vuole affrontare.

N.B.: Il file .dat può essere configurato in modo tale che la lettura dei dati avvenga da un file di Excel e che i valori della funzione obiettivo e delle variabili decisionali individuate dal risolutore siano riportate in uno stesso o differente file di Excel.

STRUTTUTARE I DATI IN IBM LOG CPLEX: INSIEMI, RANGE, ARRAY



La parte in rosso definisce un insieme, ovvero una collezione di elementi non indicizzati e non duplicati. Su tali insiemi è possibile eseguire operazioni: union, inter, diff.



Un **RANGE** definisce un intervallo ordinato di valori inter, utile per: indicizzare un array oppure specificare l'insieme dei valori che le variabili decisionali possono assumere.
La riga 11 definisce un vettore di interi di nome ***TempidiProduzione*** che ha tante componenti quanti sono gli elementi del range Componenti, nei cui valori è indicizzato.

STRUTTURARE I DATI: ESEMPIO
Delicious, azienda alimentare inglese specializzata nel settore dolciario, sta valutando la possibilità di avviare la produzione di tre nuovi prodotti. L'azienda dispone di due stabilimenti, che possono essere utilizzati entrambi per realizzare tutti i prodotti, I tempi necessari per produrre una confezione di ciascun prodotto sono evidenziati nella tabella sottostante, dove viene riportato anche il prezzo unitario.

	Tempi di Lavorazione [h]		
Prodotto	Stabilimento 1	Stabilimento 2	Profitto Unitario [€]
1	2	2	500
2	3	1	400
3	1	4	600

Per ogni stabilimento è stata individuata anche la capacità giornaliera, cioè, il numero di ore giornaliere in cui lo stabilimento può essere utilizzato. In particolare, il primo stabilimento può essere utilizzato al massimo per 10 ore al giorno, mentre la capacità giornaliera del secondo stabilimento è pari a 18 ore. Al fine di evitare un'eccessiva diversificazione della produzione e limitare i costi logistici, l'azienda ha deciso che solo uno dei due stabilimenti dovrà essere utilizzato

per la produzione e al massimo due dei tre prodotti dovranno essere messi in produzione. L'obiettivo che si vuole raggiungere è quello di massimizzare il profitto giornaliero. Bisogna imporre che solo uno dei due stabilimenti possa essere utilizzato.

Che cosa si fa allora?
È necessario individuare un insieme di prodotti e un insieme di stabilimenti. Anziché indicizzare prodotti e stabilimenti con dei numeri, si vanno ad indicizzare con delle stringhe. Come si fa? Basta scrivere tra parentesi graffe il tipo di dato di cui si ha bisogno. Per richiamare il dato, poi, gli viene dato un nome.

```
{string} Prodotti = ...;
{string} Stabilimenti = ...;
```

Il tempo di lavorazione è un array. Ci si aspetta un tempo di lavorazione sia per i prodotti che per gli stabilimenti.

```
int TempiLavorazione [Prodotti][Stabilimenti] = ...;
```

OBIETTIVO: massimizzare il profitto giornaliero
Per fare questo è necessaria una doppia sommatoria.

```
maximize sum (p in Prodotti, s in Stabilimenti) ProfittoUnitario[p] * Produzione[p][s];

subject to {

  sum (s in Stabilimenti) Attivo[s] == 1;

  sum (p in Prodotti) Selezionato[p] <= 2;

  forall (s in Stabilimenti)
    sum (p in Prodotti) Produzione [p][s] * TempiLavorazione[p][s] <= DisponibilitaOrariaStabilimento[s] * Attivo[s];

  forall (p in Prodotti)
    sum (s in Stabilimenti) Produzione[p][s] <= 1000 * Selezionato[p];
```

VARIABILI DECISIONALI E VINCOLI

$$max \sum_{i \in P} \sum_{j \in S} \pi_i x_{ij}$$

subject to:

$$\sum_{i \in P} k_i \leq 2$$

$$\sum_{j \in S} y_j = 1$$

$$\sum_{i \in P} t_{ij} x_{ij} \leq T_j y_j \; : \forall j \in S$$

$$\sum_{j \in S} x_{ij} \leq M k_i \; : \forall i \in P$$

$$k_i \in \{0,1\} \; : \forall i \in P$$

$$y_j \in \{0,1\} \; : \forall j \in S$$

$$x_{ij} \geq 0 \text{ intero} \; : \forall i \in P, j \in S$$

$$\sum_{i \in P} t_{ij} x_{ij} \leq T_j y_j \quad \forall j \in S$$

Si tratta di un vincolo che traduce simultaneamente una condizione logica ed un vincolo tecnologico.

$$\sum_{i \in P} t_{ij} x_{ij} \leq T_j \quad \forall j \in S$$

Traduzione del vincolo tecnologico sulla disponibilità oraria degli stabilimenti.

$$\sum_{i \in P} t_{ij} x_{ij} \leq M \; y_j \quad \forall j \in S$$

Traduzione della condizione logica; M è un parametro sufficientemente grande da non costituire una limitazione aggiuntiva non voluta nel caso in cui lo stabilimento j venga attivato ($yj = 1$).

SOLUZIONE

```
// solution (optimal) with objective 7200
Del prodotto P1 non viene avviata la produzione.
Del prodotto P2 viene avviata la produzione.
Del prodotto P3 non viene avviata la produzione.

Lo stabilimento S1 non viene attivato.
Lo stabilimento S2 viene attivato.

Il prodotto P2 è realizzato in 18 pezzi nello stabilimento S2
```