Esercizio 2: Quadrati

Le variabili del problema sono ovviamente le coordinate cartesiane dei centri dei quadrati. Senza perdita di generalità si può centrare il rettangolo sull'origine. Questo implica che le coordinate dei centri dei quadrati devono essere rappresentate da variabili libere.

La funzione obiettivo è non lineare ed è data da

$$\sum_{i=1}^{5} \left(x(i)^2 + y(i)^2 \right)$$

Nella sintassi di Lingo:

```
min = @sum(quadrato(i): x(i)^2 + y(i)^2);
```

I vincoli del problema devono imporre che i quadrati siano completamente racchiusi nel rettangolo. Quindi i centri dei quadrati devono giacere in un rettangolo più stretto (tanto più stretto quanto più largo è il quadrato). Dette quindi B e A la base e l'altezza del rettangolo, si hanno i seguenti 4 vincoli per ogni quadrato *i=1..5*:

$$\frac{l(i)-B}{2} \leq x(i) \leq \frac{B-l(i)}{2}$$
$$\frac{l(i)-A}{2} \leq y(i) \leq \frac{A-l(i)}{2}$$

Nella sintassi di Lingo:

```
@for(quadrato(i): x(i) >= (l(i)-B)/2);
@for(quadrato(i): x(i) <= (B-l(i))/2);
@for(quadrato(i): y(i) >= (l(i)-A)/2);
@for(quadrato(i): y(i) <= (A-l(i))/2);
```

Bisogna infine imporre la non-sovrapposizione dei quadrati. Per ogni coppia di quadrati occorre formulare 4 vincoli di non-sovrapposizione e imporre che almeno 1 dei 4 sia valido. Se tutti e 4 sono violati, allora i quadrati si sovrappongono. I 4 vincoli rappresentano le condizioni del tipo "il quadrato *i* è completamente a destra rispetto al quadrato *j*", dove "a destra" viene poi sostituito da "a sinistra", "sopra" e "sotto". E' chiaro che a due a due i vincoli sono mutualmente esclusivi. Quindi mai più di 2 possono essere verificati per ogni coppia di quadrati. Si tratta di un caso di *vincoli disgiuntivi*. Per imporre vincoli disgiuntivi si usano variabili binarie che quando assumono valore 1 rendono ridondante il vincolo. Dopodiché si può facilmente esprimere la condizione logica su quanti sono i vincoli che si ammette diventino ridondanti (nel nostro caso 3). I vincoli hanno quindi questa forma:

$$x(i) - x(j) \geq \frac{l(i)+l(j)}{2} - M * z1(i,j) \; \forall i,j : i \neq j$$

In questo caso si impone che il quadrato *i* sia tutto a destra del quadrato *j*, a meno che *z1=1*, nel qual caso il vincolo (per *M* grande abbastanza) è ridondante. Analogamente si procede negli altri tre casi, usando altre tre variabili binarie.

Nella sintassi di Lingo:

```
@for(quadrato(i): @for(quadrato(j) | j #GT# i:
        x(i)-x(j) >= (l(i)+l(j))/2 - M * z1(i,j)));
```

```
@for(quadrato(i): @for(quadrato(j) | j #GT# i:
        x(j)-x(i) >= (l(i)+l(j))/2 - M * z2(i,j)));
@for(quadrato(i): @for(quadrato(j) | j #GT# i:
        y(i)-y(j) >= (l(i)+l(j))/2 - M * z3(i,j)));
@for(quadrato(i): @for(quadrato(j) | j #GT# i:
        y(j)-y(i) >= (l(i)+l(j))/2 - M * z4(i,j)));
```

A questo punto è facile imporre che almeno uno dei 4 vincoli non sia ridondante: basta imporre che non tutte le 4 variabili binarie *z* siano pari a 1.

$$z1(i,j)+z2(i,j)+z3(i,j)+z4(i,j)\leq 3\,\forall\, i,j: i\neq j$$

Nella sintassi di Lingo:

```
@for(quadrato(i): @for(quadrato(j) | (j #GT# i):
        z1(i,j) + z2(i,j) + z3(i,j) + z4(i,j) <= 3));
```

Un valore “grande abbastanza” per *M*, in modo da garantire che i vincoli siano ridondanti quando una variabile *z* assume valore 1, è pari al doppio della massima dimensione del rettangolo, cioè 60. I vincoli infatti non possono essere violati per più di 60 unità di lunghezza (in realtà 60 è una stima prudenziale per eccesso).

Il risultante modello di programmazione non-lineare con variabili binarie è nel file Lingo QUADRATI.LG4. La soluzione, senza garanzia di ottimalità, è nel file QUADRATI.LGR.