



Progetto di Ricerca Operativa

Yu-Ji-Ho

Informazioni

Autore	Diego Stafa
E-mail	diego.stafa@studenti.unipd.it
Matricola	1226285
Data	08/06/2022
Anno	2021/22
Destinatari	Prof. Luigi De Giovanni

Sommario

Risoluzione di un problema di Programmazione Lineare riguardante la costruzione di mazzi di carte, in AMPL.

Indice

1	Scenario principale	2
1.1	Problema	2
1.2	Modello lineare	4
1.2.1	Insiemi	4
1.2.2	Parametri	4
1.2.3	Variabili	5
1.2.3.1	Variabili principali	5
1.2.3.2	Variabili di supporto	5
1.2.3.3	Variabili binarie	5
1.2.4	Funzione obiettivo	6
1.2.5	Vincoli	6
1.3	AMPL	8
1.3.1	File .dat	8
1.3.2	File .mod	10
1.3.3	File .run	19
1.3.4	Risultato	21
1.3.4.1	Considerazioni sul risultato	23
2	Scenario alternativo	23
2.1	Problema	23
2.2	Modello lineare	25
2.3	AMPL	25
2.3.1	File .dat	25
2.3.2	Risultato	27
2.3.2.1	Considerazioni sul risultato	29
3	Note finali	30

1 Scenario principale

1.1 Problema

Yu-Ji-Oh è un famoso gioco di carte collezionabili, il gioco consiste nello sfidare altre persone con dei mazzi di carte, tutto si basa nel trovare la combinazione vincente di carte durante la costruzione dei propri mazzi. Le carte possono essere di 3 tipi:

- mostro, i quali hanno un livello che varia da 1 a 5;
- magia;
- trappola.

Ogni carta si può trovare in una delle seguenti rarità:

- comune;
- rara;
- super rara;
- ultra rara.

La rarità di una carta può essere commisurata in questo modo:

- una carta rara è rara quanto 3 carte comuni;
- una carta super rara è rara quanto 5 carte comuni;
- una carta ultra rara è rara quanto 10 carte comuni.

Per la partecipazione al torneo regionale di Yu-Ji-Oh sono necessari tre mazzi di gioco e la loro costruzione deve sottostare alle seguenti regole del torneo:

- i mazzi devono avere tra le 40 e 60 carte;
- un solo mazzo può avere 60 carte;
- i mazzi devono contenere almeno 20 mostri e 10 magie;
- il livello totale dei mostri in un mazzo deve essere almeno 30.

Oltre alle regole del torneo, vogliamo costruire i nostri mazzi in modo creativo:

- i mazzi devono avere la massima rarità possibile;
- in ogni mazzo deve essere presente almeno una carta per ogni rarità;
- almeno un mazzo deve avere un mostro per ogni livello;
- il primo mazzo deve avere tante carte super rare quante il minimo numero di carte super rare tra il secondo e terzo mazzo;
- il secondo oppure il terzo mazzo devono avere al massimo 3 carte ultra rare;
- il numero di carte comuni nel primo e nel terzo mazzo deve essere uguale;

- il terzo mazzo deve contenere un numero di carte ultra rare pari alla differenza di carte super rare e comuni nel primo e secondo mazzo;
- se un mazzo ha almeno 10 mostri di livello 1 allora un'altro mazzo deve avere almeno 5 mostri di livello 5.

L'acquisto delle carte avviene in un negozio dedicato, le carte comuni sono vendute a prezzi fissi:

- i mostri costano 1€ l'uno, più 0,10€ per livello;
- le magie costano 1,5€ l'una;
- le trappole costano 0,5€ l'una.

Invece le carte in una determinata rarità hanno un valore aggiunto:

- le rare costano il 20% in più delle corrispondenti comuni;
- le super rare costano l'80% in più delle corrispondenti comuni;
- le ultra rare costano 2€ in più delle corrispondenti comuni.

1.2 Modello lineare

1.2.1 Insiemi

- **mazzo**: contiene i mazzi usati per il torneo;
- **livello**: contiene i livelli dei mostri;
- **rarita**: contiene le rarità delle carte.

1.2.2 Parametri

- **min_carte**: indica il numero minimo di carte per mazzo;
- **max_carte**: indica il numero massimo di carte per mazzo;
- **min_mostri**: indica il numero minimo di mostri per mazzo;
- **min_magie**: indica il numero minimo di magie per mazzo;
- **min_lv_tot**: indica la somma di livelli totale minima in almeno un mazzo;
- **min_mostri_lv_1**: indica il minimo numero di mostri di livello 1 in almeno un mazzo per avere *min_mostri_lv_5* in almeno un mazzo;
- **min_mostri_lv_5**: indica il minimo numero di mostri di livello 5 che devono essere presenti in almeno un mazzo quando ci sono *min_mostri_lv_1* in almeno un mazzo;
- **max_carte_ultra_secondo_terzo**: indica il massimo numero di carte ultra rare che il secondo oppure il terzo mazzo devono avere;
- **valore_rarita_{rarita}**: indica il valore di una rarità;
- **costo_mostro_{rarita, livello}**: indica il costo di un mostro in base al livello e rarità;
- **costo_magia_{rarita}**: indica il costo di una magia in base alla rarità;
- **costo_trappola_{rarita}**: indica il costo di una trappola in base alla rarità;
- **soldi_disponibili**: indica il *budget* disponibile per la costruzione dei mazzi;
- **SOMMA_LIVELLI_MASSIMA**: la somma di livelli massima possibile in un mazzo ();
- **NUM_MAZZI**: indica il numero di mazzi a disposizione;
- **NUM_LIVELLI**: indica il numero di livelli;
- **NUM_RARITA**: indica il numero di rarità.

1.2.3 Variabili

1.2.3.1 Variabili principali

- **carte_mostro**_{rarita, livello, mazzo}: indica il numero di carte mostro per: rarità, livello e mazzo;
- **carte_magia**_{rarita, mazzo}: indica il numero di carte magia per: rarità e mazzo;
- **carte_trappola**_{rarita, mazzo}: indica il numero di carte trappola per: rarità e mazzo.

1.2.3.2 Variabili di supporto

- **min_super_secondo_terzo**: indica il minimo di carte super rare tra il secondo e terzo mazzo;
- **differenza_assoluta_super_comuni**: indica la differenza, in valore assoluto, tra il numero di carte super rare del secondo mazzo e carte comuni del primo mazzo.

1.2.3.3 Variabili binarie

- **ha_60_carte**_{mazzo}: 1 se il mazzo specificato ha meno di 60 carte, 0 altrimenti;
- **lv_tot_almeno_30**_{mazzo}: 1 se il mazzo specificato ha la somma dei livelli totale minore di 30, 0 altrimenti;
- **ha_rarita**_{rarita, mazzo}: 1 se il mazzo specificato ha la rarità specificata, 0 altrimenti;
- **non_ha_tutte_rarita**_{mazzo}: 1 se il mazzo specificato non ha tutte le rarità, 0 altrimenti;
- **ha_lv**_{livello, mazzo}: 1 se il mazzo specificato ha un mostro per il livello specificato, 0 altrimenti;
- **non_ha_tutti_lv**_{mazzo}: 1 se il mazzo specificato non ha un mostro per ogni livello, 0 altrimenti;
- **ha_3_ultra_secondo_mazzo**: 1 se il secondo mazzo ha più di 3 carte ultra rare, 0 altrimenti;
- **ha_3_ultra_terzo_mazzo**: 1 se il terzo mazzo ha più di 3 carte ultra rare, 0 altrimenti;
- **ha_10_lv_1**_{mazzo}: 1 se il mazzo specificato ha almeno 10 mostri di livello 1, 0 altrimenti;
- **ha_5_lv_5**_{mazzo}: 1 se il mazzo specificato ha almeno 5 mostri di livello 5, 0 altrimenti;
- **ce_un_mazzo_con_10_lv_1**: 1 se almeno un mazzo ha 10 mostri di livello 1, 0 altrimenti;

1.2.4 Funzione obiettivo

- **massima_rarita**: la somma dei valori delle carte nei mazzi deve essere massima.

1.2.5 Vincoli

- **spesa_massima**: il costo delle carte di tutti i mazzi deve essere minore a soldi_disponibili;
- **carte_minime_{mazzo}**: ogni mazzo deve avere un numero minimo di carte;
- **carte_massime_{mazzo}**: ogni mazzo deve avere un numero massimo di carte;
- **mostri_minimi_{mazzo}**: ogni mazzo deve avere un numero minimo di mostri;
- **magie_minime_{mazzo}**: ogni mazzo deve avere un numero minimo di magie;
- **attiva_ha_60_carte_{mazzo}**: attiva la relativa variabile binaria;
- **attiva_ha_60_carte_2_{mazzo}**: attiva la relativa variabile binaria;
- **massimo_numero_mazzi_con_60_carte**: un solo mazzo può avere il massimo numero di carte;
- **attiva_lv_tot_almeno_30_{mazzo}**: attiva la relativa variabile binaria;
- **attiva_lv_tot_almeno_30_2_{mazzo}**: attiva la relativa variabile binaria;
- **massimo_numero_mazzi_con_lv_totale_30**: il livello totale dei mostri in almeno un mazzo deve essere almeno 30;
- **attiva_ha_rarita_{rarita, mazzo}**: attiva la relativa variabile binaria;
- **attiva_ha_rarita_2_{rarita, mazzo}**: attiva la relativa variabile binaria;
- **attiva_non_ha_tutte_rarita_{mazzo}**: attiva la relativa variabile binaria;
- **ogni_mazzo_con_tutte_rarita**: ogni mazzo deve contenere almeno una carta per ogni rarita;
- **attiva_ha_lv_{livello, mazzo}**: attiva la relativa variabile binaria;
- **attiva_ha_lv_2_{livello, mazzo}**: attiva la relativa variabile binaria;
- **attiva_non_ha_tutti_lv_{mazzo}**: attiva la relativa variabile binaria;
- **almeno_un_mazzo_con_tutti_lv**: almeno un mazzo deve contenere un mostro per ogni livello;
- **attiva_ha_3_ultra_secondo_mazzo**: attiva la relativa variabile binaria;
- **attiva_ha_3_ultra_secondo_mazzo_2**: attiva la relativa variabile binaria;
- **attiva_ha_3_ultra_terzo_mazzo**: attiva la relativa variabile binaria;
- **attiva_ha_3_ultra_terzo_mazzo_2**: attiva la relativa variabile binaria;
- **3_ultra_secondo_terzo**: il secondo oppure il terzo mazzo devono avere al massimo 3 carte ultra rare;
- **min_super_secondo_mazzo**: insieme a min_super_terzo_mazzo, determina il minimo di super rare tra il secondo e terzo mazzo;
- **min_super_terzo_mazzo**: insieme a min_super_secondo_mazzo, determina il minimo di super rare tra il secondo e terzo mazzo;

- **comuni_primo_mazzo_uguale_minimo:** il primo mazzo contiene tante super rare quante il minimo di super rare tra il secondo e terzo mazzo;
- **uguaglianza_comuni:** il numero di carte comuni nel il primo e terzo mazzo deve essere uguale;
- **diff_positiva:** insieme a diff_negativa, determina il valore assoluta della differenza tra il numero di carte super rare nel secondo mazzo e il numero di carte comuni nel primo mazzo;
- **diff_negativa:** insieme a diff_positiva, determina il valore assoluta della differenza tra il numero di carte super rare nel secondo mazzo e il numero di carte comuni nel primo mazzo;
- **uguaglianza_ultra:** il numero di carte ultra rare nel terzo mazzo deve essere uguale alla differenza di carte super rare nel secondo mazzo e di comuni nel primo mazzo;
- **attiva_ha_10_lv_1_mazzo:** attiva la relativa variabile binaria;
- **attiva_ha_10_lv_1_2_mazzo:** attiva la relativa variabile binaria;
- **attiva_ha_5_lv_5_mazzo:** attiva la relativa variabile binaria;
- **attiva_ha_5_lv_5_2_mazzo:** attiva la relativa variabile binaria;
- **attiva_ce_un_mazzo_con_10_lv_1:** attiva la relativa variabile binaria;
- **attiva_ce_un_mazzo_con_10_lv_1_2:** attiva la relativa variabile binaria;
- **se_10_allora_5:** se almeno un mazzo ha almeno 10 mostri di livello 1 allora almeno un mazzo deve avere almeno 5 mostri di livello 5.

1.3 AMPL

1.3.1 File .dat

```

#-----#
#-----INSIEMI-----#
#-----#

set rarita                := comune rara super ultra;
set mazzo                 := primo secondo terzo;
set livello               := 1 2 3 4 5;

#-----#
#-----PARAMETRI-----#
#-----#

# --- PARAMETRI DEL TORNEO
param min_carte            = 40;
param max_carte            = 60;
param min_mostri           = 20;
param min_magie            = 10;
param min_lv_toT           = 30;

# --- PARAMETRI CREATIVI
param min_mostri_lv_1      = 10;
param min_mostri_lv_5      = 5;
param max_carte_ultra_secondo_terzo = 3;

# --- PARAMETRI DI RARITA
param      valore_rarita :=
comune      1
rara        3
super       5
ultra       8;

# --- PARAMETRI DI PREZZO
param costo_mostro:  1  2  3  4  5  :=
comune             11 12 13 14 15
rara               13 14 16 17 18
super              20 21 23 25 27
ultra              31 32 33 34 35;

param      costo_magia :=
comune      15
rara        18
super       27
ultra       35;

param      costo_trappola :=
comune      5
rara        6
super       9
ultra      25;

param soldi_disponibili    = 150;

```

```
# --- COSTANTI MINIME DI ATTIVAZIONE

param SOMMA_LIVELLI_MASSIMA      = 300;
param NUM_MAZZI                  = 3;
param NUM_LIVELLI                 = 5;
param NUM_RARITA                  = 4;
```

1.3.2 File .mod

```
#-----#
#-----INSIEMI-----#
#-----#

# insieme delle rarita
set rarita;

# insieme dei mazzi
set mazzo;

# insieme dei livelli
set livello;

#-----#
#-----PARAMETRI-----#
#-----#

# --- PARAMETRI DEL TORNEO

# numero di carte minime per mazzo
param min_carte;

# numero di carte massime per mazzo
param max_carte;

# numero di mostri minimi per mazzo
param min_mostri;

# numero di magie minime per mazzo
param min_magie;

# somma dei livelli totali minimi in almeno un mazzo
param min_lv_tot;

# --- PARAMETRI CREATIVI

# numero minimo di mostri di livello 1
param min_mostri_lv_1;

# numero minimo di mostri di livello 5
param min_mostri_lv_5;

# numero minimo di carte ultra rare che il secondo
# o terzo mazzo devono avere
param max_carte_ultra_secondo_terzo;

# --- PARAMETRI DI RARITA

# valore di una carte in base alla rarita
param valore_rarita{rarita};
```

```
# --- PARAMETRI DI PREZZO

# costo di un mostro, dati rarita e livello
param costo_mostro{rarita, livello};

# costo di una magia, data la rarita
param costo_magia{rarita};

# costo di una trappola, data la rarita
param costo_trappola{rarita};

# soldi disponibili per l'acquisto delle carte
param soldi_disponibili;

# --- COSTANTI MINIME DI ATTIVAZIONE

param SOMMA_LIVELLI_MASSIMA;
param NUM_MAZZI;
param NUM_LIVELLI;
param NUM_RARITA;

#-----#
#-----VARIABILI-----#
#-----#

# numero di carte mostro per: rarita, livello e mazzo
var carte_mostro{rarita, livello, mazzo} >= 0 integer;

# numero di carte magia per: rarita e mazzo
var carte_magia{rarita, mazzo} >= 0 integer;

# numero di carte trappola per: rarita e mazzo
var carte_trappola{rarita, mazzo} >= 0 integer;

#-----#
#-----FUNZIONE OBIETTIVO-----#
#-----#

# la somma delle rarita in tutti i mazzi deve essere massima
maximize rarita_mazzi:
    (sum{m in mazzo, r in rarita, l in livello}
        valore_rarita[r]*carte_mostro[r, l, m])+
    (sum{m in mazzo, r in rarita}
        valore_rarita[r]*carte_magia[r, m]) +
    (sum{m in mazzo, r in rarita}
        valore_rarita[r]*carte_trappola[r, m]);

#-----#
#-----VINCOLI-----#
#-----#

# --- VINCOLI DI BUDGET
```

```

# V0: la spesa per le carte non deve superare i soldi disponibili
s.t. spesa_massima:
    (sum{m in mazzo, r in rarita, l in livello}
        costo_mostro[r,l]*carte_mostro[r, l, m])/10 +
    (sum{m in mazzo, r in rarita}
        costo_magia[r]*carte_magia[r, m])/10 +
    (sum{m in mazzo, r in rarita}
        costo_trappola[r]*carte_trappola[r, m])/10
    <=
    soldi_disponibili;

# --- VINCOLI DI TORNEO

# V1 : ogni mazzo deve avere un numero minimo di carte
s.t. carte_minime{m in mazzo}:
    (sum{l in livello, r in rarita} carte_mostro[r, l, m]) +
    (sum{r in rarita} carte_magia[r, m]) +
    (sum{r in rarita} carte_trappola[r, m])
    >=
    min_carte;

# V2 : ogni mazzo deve avere un numero massimo di carte
s.t. carte_massime{m in mazzo}:
    (sum{l in livello, r in rarita} carte_mostro[r, l, m]) +
    (sum{r in rarita} carte_magia[r, m]) +
    (sum{r in rarita} carte_trappola[r, m])
    <=
    max_carte;

# V3 : ogni mazzo deve avere un numero minimo di mostri
s.t. mostri_minimi{m in mazzo}:
    (sum{l in livello, r in rarita} carte_mostro[r, l, m])
    >=
    min_mostri;

# V4 : ogni mazzo deve avere un numero minimo di magie
s.t. magie_minime{m in mazzo}:
    (sum{r in rarita} carte_magia[r, m])
    >=
    min_magie;

# --- VINCOLI CREATIVI

# [LOGICO] V5 : un solo mazzo puo avere il massimo numero di carte

# 1 se il mazzo specificato ha meno di 60 carte, 0 altrimenti
var ha_60_carte{mazzo} binary;

s.t. attiva_ha_60_carte{m in mazzo}:
    (sum{l in livello, r in rarita} carte_mostro[r, l, m]) +
    (sum{r in rarita} carte_magia[r, m]) +
    (sum{r in rarita} carte_trappola[r, m])
    <=
    max_carte - 1 +
    max_carte*ha_60_carte[m];

```

```

s.t. attiva_ha_60_carte_2{m in mazzo}:
    (sum{l in livello, r in rarita} carte_mostro[r, l, m]) +
    (sum{r in rarita} carte_magia[r, m]) +
    (sum{r in rarita} carte_trappola[r, m])
    >=
    max_carte - 1 -
    max_carte*(1 - ha_60_carte[m]);

# soddisfa il vincolo
s.t. massimo_numero_mazzi_con_60_carte:
    (sum{m in mazzo} ha_60_carte[m])
    <=
    1;

# [LOGICO] V6 : il livello totale dei mostri in almeno un mazzo
#             deve essere almeno 30

# 1 se il mazzo specificato ha la somma dei livelli totale minore di 30, 0 altrimenti
var lv_tot_almeno_30{mazzo} binary;

s.t. attiva_lv_tot_almeno_30{m in mazzo}:
    (sum{l in livello, r in rarita} l*carte_mostro[r, l, m])
    <=
    min_lv_tot - 1 +
    max_carte*lv_tot_almeno_30[m];

s.t. attiva_lv_tot_almeno_30_2{m in mazzo}:
    (sum{l in livello, r in rarita} l*carte_mostro[r, l, m])
    >=
    min_lv_tot - 1 -
    max_carte*(1 - lv_tot_almeno_30[m]);

# soddisfa il vincolo
s.t. massimo_numero_mazzi_con_lv_totale_30:
    (sum{m in mazzo} lv_tot_almeno_30[m])
    <=
    1;

# [LOGICO] V7 : ogni mazzo deve contenere almeno una carta per ogni rarita

# 1 se il mazzo specificato ha la rarita specificata, 0 altrimenti
var ha_rarita{rarita, mazzo} binary;

# 1 se il mazzo specificato non ha tutte le rarita, 0 altrimenti
var non_ha_tutte_rarita{mazzo} binary;

s.t. attiva_ha_rarita{r in rarita, m in mazzo}:
    (sum{l in livello} carte_mostro[r, l, m]) +
    carte_magia[r, m] +
    carte_trappola[r, m]
    <=
    max_carte*ha_rarita[r, m];

```

```

s.t. attiva_ha_rarita_2{r in rarita, m in mazzo}:
    (sum{l in livello} carte_mostro[r, l, m]) +
    carte_magia[r, m] +
    carte_trappola[r, m]
    >=
    1 - max_carte*(1 - ha_rarita[r, m]);

s.t. attiva_non_ha_tutte_rarita{m in mazzo}:
    (sum{r in rarita} ha_rarita[r, m])
    >=
    NUM_RARITA - NUM_MAZZI*NUM_RARITA*non_ha_tutte_rarita[m];

# soddisfa il vincolo
s.t. ogni_mazzo_con_tutte_rarita:
    (sum{m in mazzo} non_ha_tutte_rarita[m])
    =
    0;

# [LOGICO] V9 : almeno un mazzo deve contenere un mostro per ogni livello

# 1 se il mazzo specificato ha un mostro per il livello specificato, 0 altrimenti
var ha_lv{livello, mazzo} binary;

# 1 se il mazzo specificato non ha un mostro per ogni livello, 0 altrimenti
var non_ha_tutti_lv{mazzo} binary;

s.t. attiva_ha_lv{l in livello, m in mazzo}:
    (sum{r in rarita} carte_mostro[r, l, m])
    <=
    max_carte*ha_lv[l, m]; # num lv

s.t. attiva_ha_lv_2{l in livello, m in mazzo}:
    (sum{r in rarita} carte_mostro[r, l, m])
    >=
    1 - max_carte*(1 - ha_lv[l, m]);

s.t. attiva_non_ha_tutti_lv{m in mazzo}:
    (sum{l in livello} ha_lv[l, m])
    >=
    NUM_LIVELLI - NUM_MAZZI*NUM_LIVELLI*non_ha_tutti_lv[m];

# soddisfa il vincolo
s.t. almeno_un_mazzo_con_tutti_lv:
    (sum{m in mazzo} non_ha_tutti_lv[m])
    <=
    NUM_MAZZI - 1;

# [LOGICO] V10 : il secondo oppure il terzo mazzo
# devono avere al massimo 3 carte ultra rare

# 1 se il secondo mazzo ha piu di 3 carte ultra rare, 0 altrimenti
var ha_3_ultra_secondo_mazzo binary;

```

```

# 1 se il terzo mazzo ha piu di 3 carte ultra rare, 0 altrimenti
var ha_3_ultra_terzo_mazzo binary;

s.t. attiva_ha_3_ultra_secondo_mazzo:
    (sum{l in livello} carte_mostro["ultra", l, "secondo"]) +
    carte_magia["ultra", "secondo"] +
    carte_trappola["ultra", "secondo"]
    <=
    max_carte_ultra_secondo_terzo +
    max_carte*ha_3_ultra_secondo_mazzo;

s.t. attiva_ha_3_ultra_secondo_mazzo_2:
    (sum{l in livello} carte_mostro["ultra", l, "secondo"]) +
    carte_magia["ultra", "secondo"] +
    carte_trappola["ultra", "secondo"]
    >=
    max_carte_ultra_secondo_terzo -
    max_carte*(1 - ha_3_ultra_secondo_mazzo);

s.t. attiva_ha_3_ultra_terzo_mazzo:
    (sum{l in livello} carte_mostro["ultra", l, "secondo"]) +
    carte_magia["ultra", "secondo"] +
    carte_trappola["ultra", "secondo"]
    <=
    max_carte_ultra_secondo_terzo +
    max_carte*ha_3_ultra_terzo_mazzo;

s.t. attiva_ha_3_ultra_terzo_mazzo_2:
    (sum{l in livello} carte_mostro["ultra", l, "terzo"]) +
    carte_magia["ultra", "terzo"] +
    carte_trappola["ultra", "terzo"]
    >=
    max_carte_ultra_secondo_terzo -
    max_carte*(1 - ha_3_ultra_terzo_mazzo);

# soddisfa il vincolo
s.t. 3_ultra_secondo_terzo:
    ha_3_ultra_secondo_mazzo +
    ha_3_ultra_terzo_mazzo
    <=
    1;

# V11 : il primo mazzo contiene tante super rare quante
#     il minimo di super rare tra il secondo e terzo mazzo

# contiene il minimo tra:
# - numero di carte super rare nel secondo mazzo
# - numero di carte super rare nel terzo mazzo
var min_super_secondo_terzo >= 0 integer;

s.t. min_super_secondo_mazzo:
    min_super_secondo_terzo
    <=
    (sum{l in livello} carte_mostro["super", l, "secondo"]) +

```



```

    carte_magia["super", "secondo"] +
    carte_trappola["super", "secondo"];

s.t. min_super_terzo_mazzo:
    min_super_secondo_terzo
    <=
    (sum{l in livello} carte_mostro["super", l, "terzo"]) +
    carte_magia["super", "terzo"] +
    carte_trappola["super", "terzo"];

# soddisfa il vincolo
s.t. comuni_primo_mazzo_uguale_minimo:
    (sum{l in livello} carte_mostro["super", l, "primo"]) +
    carte_magia["super", "primo"] +
    carte_trappola["super", "primo"]
    =
    min_super_secondo_terzo;

# V12 : il numero di carte comuni nel il primo
#       e terzo mazzo deve essere uguale

s.t. uguaglianza_comuni:
    (sum{l in livello} carte_mostro["comune", l, "primo"]) +
    carte_magia["comune", "primo"] +
    carte_trappola["comune", "primo"]
    =
    (sum{l in livello} carte_mostro["comune", l, "terzo"]) +
    carte_magia["comune", "terzo"] +
    carte_trappola["comune", "terzo"];

# V13 : il numero di carte ultra rare nel terzo mazzo deve essere
#       uguale alla differenza di carte super rare nel secondo mazzo
#       e di comuni nel primo mazzo

# contiene il valore assoluto della differenza tra:
# - numero di carte super rare nel secondo mazzo
# - numero di carte comuni nel primo mazzo
var differenza_assoluta_super_comuni >= 0 integer;

s.t. diff_positiva:
    differenza_assoluta_super_comuni
    >=
    (sum{l in livello} carte_mostro["super", l, "primo"]) +
    carte_magia["super", "primo"] +
    carte_trappola["super", "primo"]
    -
    (sum{l in livello} carte_mostro["comune", l, "primo"]) +
    carte_magia["comune", "primo"] +
    carte_trappola["comune", "primo"];

s.t. diff_negativa:
    differenza_assoluta_super_comuni
    >=
    -(

```

```

        (sum{l in livello} carte_mostro["super", 1, "primo"]) +
        carte_magia["super", "primo"] +
        carte_trappola["super", "primo"]
    -
        (sum{l in livello} carte_mostro["comune", 1, "primo"]) +
        carte_magia["comune", "primo"] +
        carte_trappola["comune", "primo"]
    );

# soddisfa il vincolo
s.t. uguaglianza_ultra:
    (sum{l in livello} carte_mostro["ultra", 1, "terzo"]) +
    carte_magia["ultra", "terzo"] +
    carte_trappola["ultra", "terzo"]
    =
    differenza_assoluta_super_comuni;

# [LOGICO] V14 : se almeno un mazzo ha almeno 10 mostri di livello 1
#             allora almeno un mazzo deve avere almeno 5 mostri
#             di livello 5

# 1 se il mazzo specificato ha almeno 10 mostri di livello 1, 0 altrimenti
var ha_10_lv_1{mazzo} binary;

# 1 se il mazzo specificato ha almeno 5 mostri di livello 5, 0 altrimenti
var ha_5_lv_5{mazzo} binary;

# 1 se almeno un mazzo ha 10 mostri di livello 1, 0 altrimenti
var ce_un_mazzo_con_10_lv_1 binary;

s.t. attiva_ha_10_lv_1{m in mazzo}:
    (sum{r in rarita} carte_mostro[r, 1, m])
    <=
    min_mostri_lv_1 - 1 +
    max_carte*ha_10_lv_1[m];

s.t. attiva_ha_10_lv_1_2{m in mazzo}:
    (sum{r in rarita} carte_mostro[r, 1, m])
    >=
    min_mostri_lv_1 - 1 -
    max_carte*(1 - ha_10_lv_1[m]);

s.t. attiva_ha_5_lv_5{m in mazzo}:
    (sum{r in rarita} carte_mostro[r, 5, m])
    <=
    min_mostri_lv_5 - 1 +
    max_carte*ha_5_lv_5[m];

s.t. attiva_ha_5_lv_5_2{m in mazzo}:
    (sum{r in rarita} carte_mostro[r, 5, m])
    >=
    min_mostri_lv_5 - 1 -
    max_carte*(1 - ha_5_lv_5[m]);

s.t. attiva_ce_un_mazzo_con_10_lv_1:

```

```
(sum{m in mazzo} ha_10_lv_1[m])
<=
NUM_MAZZI*ce_un_mazzo_con_10_lv_1;

s.t. attiva_ce_un_mazzo_con_10_lv_1_2:
  (sum{m in mazzo} ha_10_lv_1[m])
  >=
  - NUM_MAZZI*(1 - ce_un_mazzo_con_10_lv_1);

# soddisfa il vincolo
# spiegazione:
# - se 0 allora ci possono anche non essere 5 mostri di livello 5 nei mazzi
# - se 1 allora ci deve essere almeno un mazzo con 5 mostri di livello 5
s.t. se_10_allora_5:
  (sum{m in mazzo} ha_5_lv_5[m])
  >=
  ce_un_mazzo_con_10_lv_1;
```

1.3.3 File .run

```

reset;                                # SETUP
model yujiho.mod;                     # SELEZIONE DEL MODELLO
data yujiho.dat;                      # SELEZIONE DEI DATI
option solver cplex;                  # SELEZIONE DEL RISOLUTORE
solve;                                # AVVIO DEL RISOLUTORE

#-----#
#----VISUALIZZA FUNZIONE OBIETTIVO----#
#-----#

display rarita_mazzi;

#-----#
#----VISUALIZZA VARIABILI PRINCIPALI----#
#-----#

display {m in mazzo, l in livello, r in rarita}
    carte_mostro[r, l, m];

display {m in mazzo, r in rarita}
    carte_magia[r, m];

display {m in mazzo, r in rarita}
    carte_trappola[r, m];

#-----#
#----VISUALIZZA VARIABILI DI SUPPORTO---#
#-----#

# --- V11
display
    min_super_secondo_terzo;

# --- V13
display
    differenza_assoluta_super_comuni;

#-----#
#----VISUALIZZA VARIABILI BINARIE----#
#-----#

# --- V5
display {m in mazzo}
    ha_60_carte[m];

# --- V6
display {m in mazzo}
    lv_tot_almeno_30[m];

# --- V7
display {r in rarita, m in mazzo}
    ha_rarita[r, m];
display {m in mazzo}
    non_ha_tutte_rarita[m];

```

```
# --- V9
display {l in livello, m in mazzo}
    ha_lv[l, m];
display {m in mazzo}
    non_ha_tutti_lv[m];

# --- V10
display
    ha_3_ultra_secondo_mazzo;
display
    ha_3_ultra_terzo_mazzo;

# -- V14
display {m in mazzo}
    ha_10_lv_1[m];
display {m in mazzo}
    ha_5_lv_5[m];
display ce_un_mazzo_con_10_lv_1;
```

1.3.4 Risultato

CPLEX 20.1.0.0: optimal integer solution; objective 321

168 MIP simplex iterations

0 branch-and-bound nodes

rarita_mazzi = 321

carte_mostro[r,l,m] [primo,*,*]

: comune rara super ultra :=

1	0	13	0	0
2	0	1	0	0
3	1	0	0	0
4	1	0	0	0
5	4	0	0	0

[secondo,*,*]

: comune rara super ultra :=

1	0	20	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0
5	0	0	0	0

[terzo,*,*]

: comune rara super ultra :=

1	2	18	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0
5	0	0	0	0

;

carte_magia[r,m] :=

primo	comune	5
primo	rara	4
primo	super	0
primo	ultra	1
secondo	comune	8
secondo	rara	1
secondo	super	0
secondo	ultra	1
terzo	comune	9
terzo	rara	0
terzo	super	0
terzo	ultra	1

;

carte_trappola[r,m] :=

primo	comune	0
primo	rara	9
primo	super	1
primo	ultra	0
secondo	comune	0
secondo	rara	9
secondo	super	1
secondo	ultra	0

```

terzo  comune  0
terzo  rara    9
terzo  super   1
terzo  ultra   0
;

min_super_secondo_terzo = 1

differenza_assoluta_super_comuni = 1

ha_60_carte[m] [*] :=
  primo  0
  secondo 0
  terzo  0
;

lv_tot_almeno_30[m] [*] :=
  primo  1
  secondo 0
  terzo  0
;

ha_rarita[r,m] :=
  comune primo  1
  comune secondo 1
  comune terzo  1
  rara  primo  1
  rara  secondo 1
  rara  terzo  1
  super primo  1
  super secondo 1
  super terzo  1
  ultra primo  1
  ultra secondo 1
  ultra terzo  1
;

non_ha_tutte_rarita[m] [*] :=
  primo  0
  secondo 0
  terzo  0
;

ha_lv[l,m] :=
  1 primo  1
  1 secondo 1
  1 terzo  1
  2 primo  1
  2 secondo 0
  2 terzo  0
  3 primo  1
  3 secondo 0
  3 terzo  0
  4 primo  1
  4 secondo 0
  4 terzo  0

```

```
5 primo    1
5 secondo  0
5 terzo    0
;

non_ha_tutti_lv[m] [*] :=
    primo 0
    secondo 1
    terzo 1
;

ha_3_ultra_secondo_mazzo = 0

ha_3_ultra_terzo_mazzo = 0

ha_10_lv_1[m] [*] :=
    primo 1
    secondo 1
    terzo 1
;

ha_5_lv_5[m] [*] :=
    primo 1
    secondo 0
    terzo 0
;

ce_un_mazzo_con_10_lv_1 = 1
```

1.3.4.1 Considerazioni sul risultato

La rarità massima dei mazzi con questi vincoli è 327. Dalla stampa delle variabili utilizzate si nota come ogni vincolo sia stato rispettato. Dalla distribuzione è evidente come il risolutore abbia scelto principalmente la rarità *rara*, in quanto sebbene valga meno di altre rarità, è più economica e sottostà a molti meno vincoli rispetto alle altre.

2 Scenario alternativo

2.1 Problema

Dopo avere partecipato e vinto al torneo regionale, abbiamo ottenuto un premio che ha aumentato la nostra possibilità di spesa fino a 180 euro, perciò siamo ora interessati a partecipare al torneo nazionale di Yu-Ji-Ho. Il nuovo torneo segue regole differenti:

- per la partecipazione sono necessari 5 mazzi;
- i mazzi devono avere tra le 30 e 80 carte
- la soglia di mostri minima è stata rimossa;
- il numero minimo di magie è 20;
- il livello totale minimo in almeno un mazzo è stato alzato a 40.

In questi giorni la *Conami* ha rilasciato una nuova espansione di carte, introducendo una nuova rarità: leggenda. La rarità leggenda stravolge il vecchio sistema di rarità, modificandone prezzo e valore, ora infatti:

- una carta ultra rara è rara quanto 20 carte comuni;
- una carta leggenda è rara quanto 50 carte comuni.

Mentre per quanto riguarda i prezzi:

- i mostri leggendari costano 11 euro, più 10 centesimi per livello;
- i mostri ultra rari sono aumentati di 1 euro, mentre tutte le magie hanno perso 1 euro di valore;
- le magie leggendarie costano la metà dei mostri leggendari, senza livello;
- le trappole leggendarie costano 9 euro.

Invece le carte in una determinata rarità hanno un valore aggiunto:

- le rare costano il 20% in più delle corrispondenti comuni;
- le super rare costano l'80% in più delle corrispondenti comuni;
- le ultra rare costano 2€ in più delle corrispondenti comuni.

2.2 Modello lineare

Il modello lineare per lo scenario alternativo è invariato.

2.3 AMPL

2.3.1 File .dat

```

#-----#
#-----INSIEMI-----#
#-----#

# ALT : nuova rarita introdotta: leggenda
set rarita                                     := comune rara super ultra leggenda;

# ALT : il nuovo torneo ammette 5 mazzi
set mazzo                                     := primo secondo terzo quarto quinto;

set livello                                   := 1 2 3 4 5;

#-----#
#-----PARAMETRI-----#
#-----#

# --- PARAMETRI DEL TORNEO

# ALT : minimo numero di carte abbassato a 30
param min_carte                               = 30;

# ALT : massimo numero di carte a 80
param max_carte                               = 80;

# ALT : minimo numero di mostri rimosso
param min_mostri                              = 0;

# ALT : minimo numero di magie alzato a 20
param min_magie                              = 20;

# ALT : minimo livello totale alzato a 40
param min_lv_tot                             = 40;

# --- PARAMETRI CREATIVI

# ALT : con almeno 3 mostri di livello 1
param min_mostri_lv_1                         = 3;

# ALT : voglio almeno 3 mostri di livello 5
param min_mostri_lv_5                         = 3;

param max_carte_ultra_secondo_terzo          = 3;

# --- PARAMETRI DI RARITA

# ALT : le carte ultra aumentano di valore
param                                     valore_rarita :=
comune                                     1

```

```

rara          3
super         5
ultra        20
leggenda     50;

# --- PARAMETRI DI PREZZO

# ALT : la domanda di mostri ultra rari ne aumenta il prezzo di 1 euro
param costo_mostro:  1      2      3      4      5 :=
comune              11     12     13     14     15
rara                13     14     16     17     18
super               20     21     23     25     27
ultra               41     42     43     44     45
leggenda            110    111    112    113    114;

# ALT : l'alta offerta di magie ne diminuisce il prezzo di 1 euro
param              costo_magia :=
comune             5
rara               6
super              9
ultra              25
leggenda           55;

param              costo_trappola :=
comune             5
rara               6
super              9
ultra              25
leggenda           90;

# ALT : bonus soldi vinti al torneo
param soldi_disponibili                = 180;

# --- COSTANTI MINIME DI ATTIVAZIONE

# ALT : aggiorna somma di livelli massima
param SOMMA_LIVELLI_MASSIMA            = 400;

# ALT : aggiorna numero di mazzi
param NUM_MAZZI                        = 5;

param NUM_LIVELLI                      = 5;

param NUM_RARITA                      = 5;

```

2.3.2 Risultato

CPLEX 20.1.0.0: optimal integer solution; objective 1253

142 MIP simplex iterations

0 branch-and-bound nodes

rarita_mazzi = 1253

```
carte_mostro[r,l,m] [primo,*,*]
: comune leggenda rara super ultra :=
1  0      0      0      0      0
2  0      0      0      0      0
3  0      0      0      0      0
4  0      0      0      0      0
5  0      0      0      0      0
```

```
[quarto,*,*]
: comune leggenda rara super ultra :=
1  0      0      1      0      0
2  0      0      1      0      0
3  1      0      0      0      0
4  1      0      0      0      0
5  1      0      0      0      0
```

```
[quinto,*,*]
: comune leggenda rara super ultra :=
1  0      0      0      0      0
2  0      0      0      0      0
3  0      0      0      0      0
4  0      0      0      0      0
5  0      0      0      0      0
```

```
[secondo,*,*]
: comune leggenda rara super ultra :=
1  0      0      0      0      0
2  0      0      0      0      0
3  0      0      0      0      0
4  0      0      0      0      0
5  0      0      0      0      0
```

```
[terzo,*,*]
: comune leggenda rara super ultra :=
1  0      0      0      0      0
2  0      0      0      0      0
3  0      0      0      0      0
4  0      0      0      0      0
5  0      0      0      0      0
;
```

```
carte_magia[r,m] [*,*]
:      comune leggenda rara super ultra :=
primo  0      1      26      1      1
quarto 0      1      22      1      1
quinto 0      1      26      1      1
secondo 0     11      16      1      1
terzo  0      1      25      1      2
;
```

```

carte_trappola[r,m] [*,*]
:      comune leggenda rara super ultra :=
primo   1      0      0      0      0
quarto  0      0      0      0      0
quinto  1      0      0      0      0
secondo 1      0      0      0      0
terzo   1      0      0      0      0
;

min_super_secondo_terzo = 1

differenza_assoluta_super_comuni = 2

ha_60_carte[m] [*] :=
  primo 0
  quarto 0
  quinto 0
  secondo 0
  terzo 0
;

lv_tot_almeno_30[m] [*] :=
  primo 0
  quarto 0
  quinto 0
  secondo 0
  terzo 0
;

ha_rarita[r,m] [*,*]
:      primo quarto quinto secondo terzo :=
comune  1      1      1      1      1
leggenda 1      1      1      1      1
rara    1      1      1      1      1
super   1      1      1      1      1
ultra   1      1      1      1      1
;

non_ha_tutte_rarita[m] [*] :=
  primo 0
  quarto 0
  quinto 0
  secondo 0
  terzo 0
;

ha_lv[l,m] [*,*]
: primo quarto quinto secondo terzo :=
1  0      1      0      0      0
2  0      1      0      0      0
3  0      1      0      0      0
4  0      1      0      0      0
5  0      1      0      0      0
;

```

```
non_ha_tutti_lv[m] [*] :=
  primo 1
  quarto 0
  quinto 1
  secondo 1
  terzo 1
;

ha_3_ultra_secondo_mazzo = 0

ha_3_ultra_terzo_mazzo = 0

ha_10_lv_1[m] [*] :=
  primo 0
  quarto 0
  quinto 0
  secondo 0
  terzo 0
;

ha_5_lv_5[m] [*] :=
  primo 0
  quarto 0
  quinto 0
  secondo 0
  terzo 0
;

ce_un_mazzo_con_10_lv_1 = 0
```

2.3.2.1 Considerazioni sul risultato

La rarità massima dei mazzi con questi vincoli è 1253. Dalla stampa delle variabili utilizzate si nota come ogni vincolo sia stato rispettato. L'aumento dei mazzi dimostra chiaramente come questo influisca sulla funzione obiettivo. Dalla distribuzione delle carte si nota come il risolutore abbia scelto principalmente la rarità *rara*, sebbene la nuova rarità *legenda* valga molto di più rispetto alle altre e non abbia vincoli nuovi associati, è evidente però che il suo costo decisamente elevato la renda una cattiva scelta.

3 Note finali

- per quanto riguarda le variabili principali, si sarebbe potuto aggiungere l'insieme *tipo_carta*, ciò avrebbe permesso di sosituire le tre variabili (*carta_mostro*, *carta_magia*, *carta_trappola*) con un'unica variabile *carta*. Questo non è stato fatto in quanto i mostri contengono anche i livelli, mentre le magie e trappole no, ciò mi stava causando non pochi problemi nella stesura dei vincoli, quindi ho optato per spezzare la variabile in tre, favorendo un approccio più semplice, ma meno estensibile;
- i nomi e le descrizioni dei vincoli e delle variabili utilizzano valori numerici, ciò per motivi di chiarezza, questi valori sono in realtà parametrizzati;
- il vincolo V13 è stato lievemente modificato rispetto la proposta iniziale del problema, specificando che la differenza deve essere in valore assoluto.