

Progetto di Ricerca Operativa - Anno 2018/19

Andrea Trevisin - matr. 1144684

Padova, 29 agosto 2019

Abstract

In questo progetto viene analizzato un problema di programmazione lineare in cui un'azienda videoludica, Rubisoft Entertainment, deve cercare di massimizzare i propri guadagni a fronte delle spese necessarie per lo sviluppo di cinque diversi prodotti. In primis verrà esposto in dettaglio lo scenario principale, che sarà dunque formalizzato e modellato matematicamente. Tale modello verrà quindi riformulato nel linguaggio di programmazione AMPL e risolto mediante un ottimizzatore. Infine verranno esplorati scenari alternativi, con considerazioni su eventuali risultati significativi.

1 Problema

Rubisoft Entertainment è un piccolo studio di sviluppo videoludico americano, che sta cercando di crescere nel settore. Rubisoft ha appena stipulato un contratto con il publisher AE Games, un'azienda che gestisce sia il marketing che il rilascio dei prodotti e quindi esige che siano pronti entro una certa data. Con l'inizio dell'ultimo trimestre dell'anno, Rubisoft si trova a dover completare lo sviluppo di cinque titoli con scadenze molto stringenti. L'obiettivo di Rubisoft è di massimizzare i guadagni a fronte delle spese da sostenere.

1.1 I videogiochi

I titoli da sviluppare, la scadenza per la loro uscita e la previsione di profitto (numero di copie vendute e prezzo di ogni copia) sono i seguenti:

Videogioco	Scadenza	Copie	Prezzo
Sparatutto	2 mesi	45000	23\$
RPG	2 mesi	28000	21\$
Strategia	1 mese	15000	14\$
Avventura	3 mesi	23000	9 \$
Platform	1 mese	18000	12\$

Ogni videogioco, inoltre, richiede un certo numero di ore di lavoro specializzato in varie aree per essere completato:

Videogioco	Programmazione	Grafica	Level Design
Sparatutto	7600	2800	3200
RPG	4400	2800	4200
Strategia	1380	1600	1050
Avventura	1430	4900	1700
Platform	760	2200	3800

1.2 Team di sviluppo

Allo sviluppo di ogni prodotto viene assegnato un team di sviluppo, composto da programmatori, artisti e level designers. Il team di Rubisoft è di dimensioni ridotte, quindi l'azienda si trova costretta a fare outsourcing ed ingaggiare professionisti costosi per poter completare il lavoro. Rubisoft al momento dispone del seguente numero di dipendenti per ciascun ruolo:

Ruolo	Numero
Programmatore	7
Artista	3
Level Designer	6

Gli stipendi mensili per un professionista per ciascun ruolo sono i seguenti:

Ruolo	Stipendio
Programmatore	4760
Artista	3600
Level Designer	4950

Un professionista può essere ingaggiato solo per l'intera durata del progetto. Un dipendente o un professionista lavorano, in media, 160 ore al mese. Se un

team di sviluppo è composto da più di 25 persone, lavorerà peggio a causa di conflitti interni, abbassando del 14% l'ammontare di ore di lavoro utile da parte di ogni membro. Questo può essere prevenuto dalla presenza di almeno 5 dipendenti nel team, grazie alla loro esperienza nel lavorare assieme.

1.3 Motore grafico

Per ogni prodotto Rubisoft può scegliere di impiegare uno tra 2 motori grafici: Harmony, un motore grafico con funzioni base, o Real Engine, un prodotto professionale più costoso ed efficiente che, grazie alle sue features, riduce il carico nelle aree di programmazione e level design del 12%. All'acquisto deve essere scelto il tipo di licenza: a costo fisso (è necessario acquistare una licenza per ogni programmatore e level designer) o a royalty (si compra una licenza unica per l'intero team che segue il progetto, pagando una certa percentuale del ricavo per ogni copia venduta). Per ogni videogioco *deve* essere scelto un motore grafico, che corrisponde all'acquisto di almeno una licenza (di qualunque tipo).

Motore grafico	Prezzo fisso	Royalty
Harmony	439\$	2%
Real Engine	839\$	5%

1.4 Incentivi

Aumentando le ore di lavoro del 33% nell'area di programmazione e del 45% nell'area grafica è possibile scegliere di aggiungere micro-transazioni ad un videogioco; ogni copia venduta di un gioco con micro-transazioni frutterà, in media, 3 dollari in più (che non vanno contati nel calcolo della royalty per il motore grafico).

2 Modellazione

2.1 Definizione degli insiemi

Gli insiemi considerati sono i seguenti:

- $V = \{Fps, Rpg, Str, Adv, Plt\}$: insieme dei videogiochi
- $I = \{Dip, Pro\}$: insieme dei tipi di impiego

- $R = \{Pr, Ar, Ld\}$: insieme dei ruoli nei team di sviluppo
- $E = \{Ha, Re\}$: insieme dei motori grafici

2.2 Definizione delle variabili decisionali

Le variabili decisionali considerate sono le seguenti:

- $X_{i,v}$: numero di programmatori con impiego $i \in I$ assegnati al videogioco $v \in V$
- $Y_{i,v}$: numero di artisti con impiego $i \in I$ assegnati al videogioco $v \in V$
- $Z_{i,v}$: numero di level designer con impiego $i \in I$ assegnati al videogioco $v \in V$
- $P_{e,v}$: variabile logica che indica la scelta di utilizzare il motore grafico $e \in E$ per lo sviluppo del videogioco $v \in V$
- $W_{e,v}$: numero di licenze a prezzo fisso acquistate per il motore grafico $e \in E$ per lo sviluppo del videogioco $v \in V$
- $Q_{e,v}$: variabile logica che indica l'acquisto di una licenza a royalty per il motore grafico $e \in E$ per lo sviluppo del videogioco $v \in V$
- S_v : variabile logica che indica la presenza conflitti nel team di sviluppo per il videogioco $v \in V$
- T_v : variabile logica che indica il superamento della dimensione critica del team di sviluppo per il videogioco $v \in V$
- U_v : variabile logica che indica la presenza di un numero sufficiente di dipendenti nel team di sviluppo per il videogioco $v \in V$
- O_v : variabile logica che indica l'intenzione di implementare micro-transazioni nel videogioco $v \in V$

2.3 Definizione dei parametri

I parametri considerati sono i seguenti:

- $hn_{r,v}$: numero di ore di lavoro nell'area pertinente al ruolo $r \in R$ necessarie per il videogioco $v \in V$
- dl_v : scadenza in mesi per il completamento del videogioco $v \in V$
- qc_v : numero in copie che si prevede di vendere per il videogioco $v \in V$
- cc_v : prezzo in dollari di una copia videogioco $v \in V$
- d_r : numero di dipendenti con ruolo $r \in R$
- st_r : stipendio mensile in dollari per un professionista con ruolo $r \in R$
- hm : quantità in ore di lavoro che una persona svolge in un mese
- mt : numero minimo di membri in un team tale che si verifichino conflitti interni
- md : numero minimo di dipendenti in un team tale che si evitino conflitti interni in un team
- phw : percentuale di ore di lavoro utile sprecate da conflitti interni ad un team
- phs : percentuale di riduzione delle ore di lavoro necessarie con l'utilizzo del motore grafico più efficiente
- pap : percentuale ore di programmazione aggiuntive per implementare micro-transazioni in un videogame
- pag : percentuale ore di grafica aggiuntive per implementare micro-transazioni in un videogame
- fl_e : costo in dollari di una licenza a prezzo fisso per il motore grafico $e \in E$
- rl_e : costo percentuale sul ricavo in dollari di una licenza a royalty per il motore grafico $e \in E$

- av : valore aggiunto in dollari per ogni copia di un videogioco con micro-transazioni
- N : parametro arbitrariamente grande

2.4 Funzione obiettivo

$$\begin{aligned} \max \sum_{v \in V} qc_v \cdot (cc_v + av \cdot O_v) - (X_{Pro,v} \cdot st_{Pr} + Y_{Pro,v} \cdot s.t._{Ar} + Z_{Pro,v} \cdot st_{Ld}) \\ - \sum_{v \in V, e \in E} fl_e \cdot W_{e,v} + rl_e \cdot Q_{e,v} \cdot qc_v \cdot cc_v \end{aligned}$$

2.5 Vincoli

Nella modellazione di questo problema mi sono scontrato con un paio di situazioni interessanti, che verranno commentate vicino ai vincoli interessati. In generale, valgono i seguenti vincoli:

Programmatori dipendenti dell'azienda disponibili

$$\sum_{v \in V} X_{Dip,v} \leq d_{Pr}$$

Artisti dipendenti dell'azienda disponibili

$$\sum_{v \in V} Y_{Dip,v} \leq d_{Ar}$$

Level designer dipendenti dell'azienda disponibili

$$\sum_{v \in V} Z_{Dip,v} \leq d_{Ld}$$

Per ogni videogioco ($\forall v \in V$;) valgono i seguenti vincoli:

Numero di programmatori per soddisfare le ore di programmazione necessarie (e attivazione variabile logica)

$$\sum_{i \in I} X_{i,v} \cdot hm \cdot dl_v \geq hn_{Pr,v} \cdot (1 + phw \cdot S_v - phs \cdot P_{Re,v} + pap \cdot O_v)$$

Numero di artisti per soddisfare le ore di grafica necessarie

$$\sum_{i \in I} Y_{i,v} \cdot hm \cdot dl_v \geq hn_{Ar,v} \cdot (1 + phw \cdot S_v + pag \cdot O_v)$$

Numero di level designer per soddisfare le ore di level design necessarie

$$\sum_{i \in I} Z_{i,v} \cdot hm \cdot dl_v \geq hn_{Ld,v} \cdot (1 + phw \cdot S_v + phs \cdot P_{Re,v})$$

Assumiamo una condizione del tipo $y = 1$ sse $x \geq k$, come quelle per il raggiungimento della soglia critica del team o del numero di dipendenti per evitare conflitti interni; per mantenere la linearità del problema vanno definiti due vincoli distinti (dato N sufficientemente grande, e presumendo variabili intere):

1. $x - k + 1 \leq N \cdot y$ garantisce che quando x raggiunge la soglia allora y deve necessariamente essere uguale a 1, in quanto a sinistra della disequazione avremmo una quantità positiva.
2. $y \cdot k \leq x$ garantisce che quando x non raggiunge la soglia allora y deve necessariamente essere uguale a 0, in quanto la disequazione non sarebbe rispettata.

Dimensione critica del team di sviluppo (e attivazione variabile logica)

1. $(\sum_{i \in I} X_{i,v} + Y_{i,v} + Z_{i,v}) - mt + 1 \leq N \cdot T_v$
2. $T_v \cdot mt \leq \sum_{i \in I} X_{i,v} + Y_{i,v} + Z_{i,v}$

Presenza sufficiente di dipendenti nel team per risolvere i conflitti (e attivazione variabile logica)

1. $X_{Dip,v} + Y_{Dip,v} + Z_{Dip,v} - md + 1 \leq N \cdot U_v$
2. $U_v \cdot md \leq X_{Dip,v} + Y_{Dip,v} + Z_{Dip,v}$

Presenza di conflitti interni al team di sviluppo (e attivazione variabile logica)

1. $S_v \leq T_v$
2. $S_v \leq 1 - U_v$
3. $S_v \geq T_v - U_v$: copre il caso limite in cui si hanno $T = 1$ e $U = 0$ (quindi si avrebbe $S \leq 1$, un vincolo troppo lasco), imponendo $S \geq 1$ per assicurare il valore corretto di $S = 1$.

La scelta del motore grafico e del tipo di licenza è stata interessante da modellare. Per prima cosa bisogna verificare che per un videogioco sia stato scelto uno e un solo motore grafico; quindi si verifica che se il motore grafico è stato scelto, allora non è possibile non aver acquistato licenze; infine si verifica che se è stata acquistata una licenza a royalty non siano state acquistate licenze a costo fisso e viceversa.

Esclusività del motore grafico utilizzato (e attivazione variabili logiche)

$$\sum_{e \in E} P_{e,v} = 1$$

Acquisto di almeno una licenza se e solo il motore grafico viene utilizzato

1. $\forall e \in E : Q[e, v] + W[e, v] \leq N \cdot P[e, v]$: verifica che se non viene utilizzato il motore grafico non vengano acquistate licenze.
2. $\forall e \in E : Q[e, v] + W[e, v] \geq 1 - N \cdot (1 - P[e, v])$: verifica che se viene utilizzato il motore grafico venga acquistato almeno un tipo di licenza.

Esclusività del tipo di licenza acquistata

$$\forall e \in E : W_{e,v} \leq N \cdot (1 - Q_{e,v})$$

Numero di licenze singole da acquistare

$$\forall e \in E : W_{e,v} \geq (\sum_{i \in I} X_{i,v} + Z_{i,v}) - N \cdot (1 - P_{e,v} + Q_{e,v})$$

2.6 Domini

- $X_{i,v}, Y_{i,v}, Z_{i,v}, W_{e,v} \in \mathbb{N} \wedge X_{i,v}, Y_{i,v}, Z_{i,v}, W_{e,v} \geq 0$
- $P_{e,v}, Q_{e,v}, S_v, T_v, U_v, O_v \in 0, 1$

3 Risoluzione in AMPL

Il modello di programmazione lineare esposto è quindi stato tradotto in linguaggio AMPL e risolto con il risolutore **cplex**. La traduzione in linguaggio AMPL prevede di creare 3 file distinti: un file **.mod** contenente la descrizione del modello in esame, un file **.dat** contenente i dati su cui effettuare i calcoli, ed infine un file **.run** con le istruzioni per la risoluzione del problema.

Il codice è disponibile in allegato alla presente relazione; di seguito viene ri-

portato un estratto interessante del file .mod, contenente la funzione obiettivo e la definizione dei vincoli:

```
#Funzione obiettivo
maximize GuadagnoTot : sum{v in V}(qc[v] * (cc[v] + av
    * O[v]) - (X["Pro",v] * st["Pr"] + Y["Pro",v] *
    st["Ar"] + Z["Pro",v] * st["Ld"])) - sum{e in E,v
    in V}(fl[e] * W[e,v] + rl[e] * Q[e,v] * qc[v] *
    cc[v]);

#Vincoli
5 subject to programmatoriDipendenti:
    sum{v in V} X["Dip",v] <= d["Pr"];

    subject to artistiDipendenti :
    sum{v in V} Y["Dip",v] <= d["Ar"];

10 subject to leveledesignerDipendenti :
    sum{v in V} Z["Dip",v] <= d["Ld"];

    subject to oreProgrammazione{v in V} :
15 sum{i in I} X[i,v] * hm * dl[v] >= hn["Pr",v] * (1 +
    phw * S[v] - phs * P["Re",v] + pap * O[v]);

    subject to oreGrafica{v in V}:
    sum{i in I} Y[i,v] * hm * dl[v] >= hn["Ar",v] * (1 +
    phw * S[v] + pag * O[v]);

20 subject to oreLevelDesign{v in V} :
    sum{i in I} Z[i,v] * hm * dl[v] >= hn["Ld",v] * (1 +
    phw * S[v] - phs * P["Re",v]);

    # se ho abbastanza dipendenti forza U a 1
    subject to dipendentiNelTeam1{v in V} :
25 (X["Dip",v] + Y["Dip",v] + Z["Dip",v]) - md + 1 <= N *
    U[v];

    # se non ho abbastanza dipendenti forza U a 0
```

```

    subject to dipendentiNelTeam2{v in V} :
    U[v] * md <= (X["Dip",v] + Y["Dip",v] + Z["Dip",v]);

30 # se ho raggiunto la dimensione critica forza T a 1
    subject to dimensioneCritical1{v in V} :
    sum{i in I} (X[i,v] + Y[i,v] + Z[i,v]) - mt + 1 <= N *
        T[v];

35 # se non ho raggiunto la dimensione critica forza T a
    0
    subject to dimensioneCritica2{v in V} :
    T[v] * mt <= sum{i in I} (X[i,v] + Y[i,v] + Z[i,v]);

    subject to presenzaConflitti1{v in V} :
40 S[v] <= T[v];

    subject to presenzaConflitti2{v in V} :
    S[v] >= T[v] - U[v];

45 subject to presenzaConflitti3{v in V} :
    S[v] <= (1 - U[v]);

    subject to esclusivitaMotoreGrafico{v in V}:
50 sum{e in E} P[e,v] = 1;

    # forzo ad essere 0 se P è 0
    subject to acquistoLicenza1{v in V,e in E}:
    Q[e,v] + W[e,v] <= N * P[e,v];

55 # forzo ad essere almeno 1 se P è 1
    subject to acquistoLicenza2{v in V,e in E}:
    Q[e,v] + W[e,v] >= 1 - N * (1 - P[e,v]);

60 subject to sceltaTipoLicenza{v in V,e in E}:
    W[e,v] <= N * (1 - Q[e,v]);

    subject to numeroMinimoLicenze{v in V,e in E}:

```

$$W[e, v] \geq \text{sum}\{i \text{ in } I\} (X[i, v] + Z[i, v]) - N * (1 - P[e, v] + Q[e, v]) ;$$

3.1 Risultati e considerazioni

Di seguito vengono riportati i risultati dell'elaborazione nello scenario esposto:

	GuadagnoTot = 1769560	
	Programmatori	
	X :=	
5	Dip Adv	0
	Dip Fps	2
	Dip Plt	5
	Dip Rpg	0
	Dip Str	0
10	Pro Adv	4
	Pro Fps	27
	Pro Plt	1
	Pro Rpg	17
	Pro Str	8
15	;	
	Artisti	
	Y :=	
	Dip Adv	0
20	Dip Fps	3
	Dip Plt	0
	Dip Rpg	0
	Dip Str	0
	Pro Adv	15
25	Pro Fps	11
	Pro Plt	20
	Pro Rpg	13
	Pro Str	10
	;	
30		

	Level Designers	
	Z :=	
	Dip Adv	1
	Dip Fps	0
35	Dip Plt	0
	Dip Rpg	5
	Dip Str	0
	Pro Adv	3
	Pro Fps	9
40	Pro Plt	21
	Pro Rpg	7
	Pro Str	6
	;	
45	Motore grafico	
	P :=	
	Ha Adv	1
	Ha Fps	0
	Ha Plt	0
50	Ha Rpg	0
	Ha Str	0
	Re Adv	0
	Re Fps	1
	Re Plt	1
55	Re Rpg	1
	Re Str	1
	;	
	Licenze a costo fisso	
60	W :=	
	Ha Adv	8
	Ha Fps	0
	Ha Plt	0
	Ha Rpg	0
65	Ha Str	0
	Re Adv	0
	Re Fps	38
	Re Plt	0

```

70   Re Rpg    29
    Re Str     0
    ;

    Licenze a royalty
    Q :=
75   Ha Adv     0
    Ha Fps     0
    Ha Plt     0
    Ha Rpg     0
    Ha Str     0
80   Re Adv     0
    Re Fps     0
    Re Plt     1
    Re Rpg     0
    Re Str     1
85   ;

    Team con conflitti interni
    S [*] :=
90   Adv     0
    Fps     0
    Plt     0
    Rpg     0
    Str     0
    ;

95   Team che superano la dimensione critica
    T [*] :=
    Adv     0
    Fps     1
100  Plt     1
    Rpg     1
    Str     0
    ;

105  Team con abbastanza dipendenti
    U [*] :=

```

	Adv	0
	Fps	1
	Plt	1
110	Rpg	1
	Str	0
	;	
	Videogiochi con micro-transazioni	
115	O [*] :=	
	Adv	1
	Fps	1
	Plt	1
	Rpg	1
120	Str	0
	;	

Il guadagno totale di Rubisoft è di 1769560\$. Consideriamo i risultati per i vari videogame:

- **Sparatutto:** questo gioco era uno dei più impegnativi e il team di sviluppo ha ovviamente superato la dimensione critica; tuttavia vi sono stati assegnati abbastanza dipendenti per evitare conflitti interni. È stato scelto di usare il motore grafico più costoso comprando licenze a costo fisso, il che è intuibile dato che è il videogioco con il ricavo - e quindi la royalty - più alti. Infine è stato scelto di implementare le micro-transazioni nonostante il notevole carico di lavoro aggiuntivo, scelta riconducibile all'alto numero di copie vendute.
- **RPG:** altro gioco molto impegnativo, per cui valgono tutte le considerazioni esposte per il videogame precedente; è interessante notare che vengono impiegati 5 dei 6 level designer dipendenti Rubisoft.
- **Strategia:** per questo gioco viene impiegato un team che non supera la dimensione critica di 25 persone, ed è quindi naturale che non gli vengano assegnati dipendenti Rubisoft. Nonostante ciò il carico di lavoro è abbastanza alto da portare a scegliere il motore grafico più costoso, questa volta con licenza a royalty - indicando un ricavo abbastanza piccolo da preferirla all'acquisto di licenze a costo fisso. nonostante le già menzionate dimensioni ridotte del team di sviluppo. Si tratta dell'unico gioco per cui è stato scelto di non implementare le micro-transazioni.

- **Avventura:** anche qui il team non supera i 25 membri, e conta 1 level designer di Rubisoft (ci sono, in tutto, 16 dipendenti Rubisoft e 3 team che raggiungono la dimensione critica: ad ognuno di questi vengono assegnati 5 dipendenti, lasciandone uno "spaiato"). È il solo gioco in cui viene impiegato il motore grafico più economico, con licenze a costo fisso.
- **Platform:** 5 dei 7 programmatori dipendenti di Rubisoft sono stati assegnati questo videogioco, che richiedeva il minor numero di ore di programmazione, portando ad assumere un solo professionista in tale ruolo. Anche in questo caso vengono implementate le micro-transazioni, e viene impiegato Real Engine con licenza a royalty.

Osserviamo quindi dei trend in favore dell'implementazione di microtransazioni e dell'utilizzo del motore grafico più costoso ma più efficiente, oltre ovviamente all'equa divisione dei dipendenti nei team di sviluppo più corposi.

3.2 Scenari alternativi

Consideriamo ora alcuni scenari alternativi all'originale in cui verranno variati dei singoli parametri e osserviamo come variano i risultati rispetto ai trend esposti in precedenza:

- **Abbassando l'efficienza del motore grafico Real Engine al 7%** si osserva come rimanga in uso solo per lo sviluppo del videogame platform (con royalty), mentre per gli altri giochi viene utilizzato Harmony (con licenze individuali per il gioco d'avventura e il gioco sparatutto, e royalty per il gioco RPG e quello di strategia).
- **Abbassando il ricavo delle micro-transazioni a 1 dollaro per copia** si osserva che l'unico gioco a continuare ad implementarle è il gioco d'avventura (smettono di essere implementate per i giochi sparatutto, RPG e platform), dato che è quello con la scadenza più lontana e soprattutto vende un moderato numero di copie ad un prezzo abbastanza basso - e le micro-transazioni sono un incremento *flat* al valore di ogni copia.
- **Raddoppiando la royalty per l'utilizzo di Real Engine** si osserva che non viene più utilizzato per il gioco di strategia, ma continua ad essere scelto per lo sviluppo del videogioco platform.

- **Abbassando la dimensione critica del team di sviluppo a 20 persone** si osserva che i dipendenti Rubisoft sono ripartiti allo stesso modo, lasciando 2 team con conflitti interni (assegnati rispettivamente al videogioco platform e a quello di strategia). In questo scenario il videogioco di strategia implementa le microtransazioni, e per il videogioco di avventura Rubisoft sceglie di acquistare licenze individuali per il motore grafico più efficiente - portando Real Engine ad essere utilizzato per lo sviluppo di tutti i videogiochi, scelta logica data la necessità di abbassare il carico di lavoro.
- **Abbassando i programmatori dipendenti Rubisoft a 5**, uno dei team di sviluppo che superano la dimensione critica rimane “scoperto” (non ci sono abbastanza dipendenti per ogni team), e corrisponde a quello assegnato al videogioco RPG. In questo caso per tale gioco viene scelto il motore grafico Harmony con licenza a royalty (rispetto a Real Engine con licenze individuali).

In allegato vengono forniti i relativi file **.dat**.