

POLITECNICO DI TORINO

Corso di Laurea Magistrale
in Ingegneria Gestionale

Tesi di Laurea Magistrale

Simulare un portafoglio di progetti: modello ed implementazione



**Politecnico
di Torino**

Relatore

Prof. Alberto De Marco

Correlatore

Ing. Massimo Rebuglio

Candidato

Francesco Ciotta

Anno Accademico 2023-2024

Sommario

Questa tesi esplora lo sviluppo e l'implementazione di un modello innovativo per la gestione di un portafoglio di progetti. Nel primo capitolo, dopo un'introduzione dettagliata al project management e ai sistemi informativi di gestione dei progetti (PMIS), vengono discussi i principali modelli esistenti in letteratura, evidenziando le sfide legate alla gestione simultanea di più progetti con risorse condivise. Nel secondo capitolo, il lavoro propone un nuovo modello che consente ai project manager di simulare metriche di progetto e adattarsi dinamicamente alle variazioni dei parametri, con l'obiettivo di ottimizzare tempi e costi; qui viene presentata la struttura del modello, l'algoritmo evolutivo utilizzato per la schedulazione e i dati utilizzati nelle simulazioni. Nel terzo capitolo, vengono presentati i risultati delle simulazioni che dimostrano l'efficacia del modello nel migliorare la schedulazione e la gestione delle risorse, riducendo i costi e incrementando l'efficienza complessiva del progetto. La tesi conclude con una discussione sui risultati ottenuti e sulle possibili estensioni future del modello.

Ringraziamenti

Concludere questo intenso percorso rappresenta un traguardo importante per me, che non sarebbe stato possibile senza l'aiuto, il sostegno e l'ispirazione di molte persone a cui desidero esprimere i miei più profondi ringraziamenti.

In primo luogo, desidero ringraziare il mio relatore, Alberto De Marco, e il mio correlatore, Massimo Rebuglio, per la loro guida, la disponibilità costante e i consigli preziosi. Il loro supporto e le loro competenze sono state fondamentali per la realizzazione di questo lavoro.

Ringrazio la mia famiglia, per il sostegno incondizionato che hanno dimostrato in questi anni. La vostra presenza e i vostri consigli sono stati fondamentali per arrivare fin qui.

Un grazie va anche a tutti i miei amici e colleghi, che hanno condiviso con me questo viaggio. Grazie per il vostro supporto e per i momenti di leggerezza che hanno reso questi anni indimenticabili.

Indice

Elenco delle tabelle	5
Elenco delle figure	6
1 Introduzione	7
1.1 Il Project Management	7
1.2 Project Management Information System	8
1.2.1 Pianificazione e programmazione	9
1.2.2 Gestione delle risorse	9
1.2.3 Monitoraggio e controllo	9
1.2.4 Comunicazione	9
1.2.5 Gestione dei rischi	9
1.2.6 Reportistica e analisi	10
1.3 Modelli di PMIS presenti in letteratura	10
1.4 Programmazione non lineare	13
1.5 Algoritmi di ottimizzazione metaeuristica	14
1.6 Gap Statement	15
1.7 Obiettivo	17
2 Approccio	19
2.1 Modello	19
2.1.1 Diagramma e notazioni utilizzate	20
2.1.2 Descrizione delle classi	20
2.2 Dati	23
2.3 Implementazione	25
2.3.1 Programmazione iniziale	25
2.3.2 Algoritmo evolutivo	26
3 Risultati	31
3.1 Mappatura genotipo-fenotipo e boundaries	31
3.2 Prima simulazione	32

3.2.1	Schedulazione iniziale	32
3.2.2	Applicazione del modello	33
3.3	Seconda simulazione	37
3.3.1	Schedulazione iniziale	38
3.3.2	Applicazione del modello	40
4	Conclusioni	46

Elenco delle tabelle

2.1	Caratteristiche dei dataset utilizzati	23
3.1	Risultati schedulazione iniziale - PRG_1	33
3.2	Misure del carico delle risorse di PRG_1 con la schedulazione iniziale	34
3.3	Risultati del modello su PRG_1	36
3.4	Misure del carico delle risorse di PRG_1 con l'applicazione del modello	37
3.5	Risultati schedulazione iniziale - PRG_2	39
3.6	Misure del carico delle risorse di PRG_2 con la schedulazione iniziale	39
3.7	Risultati del modello su PRG_2	42
3.8	Misure del carico delle risorse di PRG_2 con l'applicazione del modello	42

Elenco delle figure

2.1	UML class diagram	22
2.2	Schema a blocchi dell'algoritmo Differential Evolution	28
3.1	Gantt del programma PRG_1 con la schedulazione iniziale	33
3.2	Carico delle risorse di PRG_1 con la schedulazione iniziale	34
3.3	Gantt del programma PRG_1 con l'applicazione del modello	36
3.4	Andamento della funzione obiettivo - PRG_1	37
3.5	Carico delle risorse di PRG_1 con l'applicazione del mdoello	38
3.6	Gantt del programma PRG_2 con la schedulazione iniziale	40
3.7	Carico delle risorse di PRG_2 con la schedulazione iniziale	41
3.8	Gantt del programma PRG_2 con l'applicazione del modello	43
3.9	Carico delle risorse di PRG_2 con l'applicazione del modello	44
3.10	Andamento della funzione obiettivo - PRG_2	45

Capitolo 1

Introduzione

1.1 Il Project Management

Il Project Management Institute (PMI) definisce il project management come *"l'applicazione di conoscenze, attitudini, strumenti e tecniche alle attività di un progetto al fine di conseguirne gli obiettivi"*. Secondo il socio fondatore del PMI, Russell D. Archibald, *"il project management è la gestione sistemica di un'impresa complessa, unica, di durata limitata, rivolta al raggiungimento di un obiettivo chiaramente predeterminato, mediante un processo continuo di pianificazione e controllo di risorse differenziate e con vincoli interdipendenti di costi-tempi-qualità"*. Queste definizioni, sebbene non siano le uniche presenti in letteratura, ci permettono di comprendere il project management come una disciplina che coinvolge la pianificazione, l'organizzazione e la gestione delle risorse per raggiungere obiettivi specifici e criteri di successo prefissati. È un processo fondamentale per garantire che i progetti vengano completati nei tempi previsti, rimanendo entro il budget e rispettando i requisiti di qualità.

Le origini del project management possono essere ricondotte a pratiche antiche, tuttavia la disciplina ha preso forma come campo di studio e pratica professionale a partire dal XX secolo. L'introduzione di metodologie e strumenti specifici ha rivoluzionato il modo in cui i progetti vengono gestiti. Tra i più noti vi sono il Metodo del Percorso Critico (CPM), sviluppato negli anni '50 per ottimizzare la sequenza delle attività in un progetto, il Diagramma di Gantt, introdotto negli anni '20 come strumento di visualizzazione delle tempistiche di progetto, e il Project Evaluation and Review Technique (PERT), utilizzato per la gestione di progetti complessi e incerti.

L'elemento centrale del project management è il progetto, definito dal PMI come un'iniziativa o uno sforzo temporaneo intrapreso per creare un prodotto, servizio o risultato unico. A differenza delle operazioni di routine, che sono continue e

ripetitive, i progetti sono unici e hanno un inizio e una fine ben definiti. La temporaneità e l'unicità sono quindi le caratteristiche distintive dei progetti. Gli elementi fondamentali che costituiscono un progetto includono lo scopo e i suoi vincoli, in termini di tempi, costi e qualità. Il successo di un progetto dipende dalla capacità di bilanciare questi vincoli in modo efficace, garantendo che i risultati finali soddisfino le aspettative degli stakeholder.

L'implementazione efficace del project management può portare a significativi vantaggi competitivi per le organizzazioni. Migliorando la capacità di adattarsi ai cambiamenti del mercato e alle esigenze dei clienti, le aziende possono ottimizzare l'uso delle risorse e raggiungere gli obiettivi strategici con maggiore efficienza. Ad esempio, una gestione rigorosa dei progetti può ridurre i tempi di consegna, migliorare la qualità dei prodotti o servizi e aumentare la soddisfazione degli stakeholder. Inoltre, un approccio strutturato al project management consente di identificare e gestire i rischi in modo proattivo, minimizzando l'impatto delle variabili inattese che possono influenzare negativamente il progetto.

La complessità e l'incertezza intrinseche ai progetti richiedono competenze specifiche e un approccio metodico per gestire le sfide che possono emergere lungo il percorso. La gestione del rischio è un aspetto cruciale del project management, poiché permette di anticipare problemi potenziali e sviluppare piani di mitigazione appropriati. Inoltre, la comunicazione efficace tra i membri del team di progetto e gli stakeholder è fondamentale per garantire che tutte le parti interessate siano allineate sugli obiettivi e sugli avanzamenti del progetto.

In conclusione, il project management non è solo una serie di tecniche e strumenti, ma una vera e propria filosofia di gestione che mira a trasformare le idee in realtà concrete. Con una pianificazione accurata, un'organizzazione efficiente e un controllo rigoroso, i project manager possono guidare i loro team verso il successo, realizzando progetti che portano valore e innovazione alle loro organizzazioni.

1.2 Project Management Information System

Nel contesto del project management un ruolo cruciale è svolto dagli sistemi informativi. In particolare, i Project Management Information System (PMIS) rappresentano una categoria specifica di sistemi informativi che sono progettati per supportare e migliorare il processo di gestione dei progetti. Questi sistemi sono fondamentali per le fasi di pianificazione, esecuzione e controllo, e offrono una gamma di funzionalità che facilitano la gestione complessiva del ciclo di vita di un progetto. Il compito dei PMIS è quindi quello di facilitare la raccolta, l'analisi e la distribuzione delle informazioni necessarie per prendere decisioni informate e tempestive. Consentono una gestione efficiente delle risorse, il monitoraggio delle

tempistiche e dei budget, nonché la comunicazione e la collaborazione tra i membri del team e gli stakeholder.

1.2.1 Pianificazione e programmazione

I PMIS aiutano i project manager nella creazione di piani di progetto dettagliati. Questi sistemi offrono strumenti per la definizione degli obiettivi, la pianificazione delle attività, la stima delle risorse e la previsione delle tempistiche. Attraverso moduli di Gantt e di diagrammi di rete, i PMIS permettono di visualizzare le interdipendenze tra le attività e di identificare il percorso critico, essenziale per garantire che il progetto sia completato nei tempi stabiliti.

1.2.2 Gestione delle risorse

Uno dei ruoli principali dei PMIS è la gestione delle risorse. Questi sistemi consentono ai project manager di allocare le risorse umane e materiali in modo efficiente, monitorando la disponibilità e l'utilizzo delle stesse. La possibilità di vedere in tempo reale lo stato delle risorse aiuta a prevenire sovraccarichi di lavoro e garantisce che le risorse siano impiegate in maniera ottimale.

1.2.3 Monitoraggio e controllo

I PMIS forniscono strumenti avanzati per il monitoraggio e il controllo dei progetti. Grazie a dashboard interattivi e report personalizzabili, i project manager possono tenere traccia dell'avanzamento delle attività, confrontare i progressi con i piani originali e identificare tempestivamente eventuali deviazioni. Questi sistemi facilitano anche il controllo dei costi, permettendo di monitorare le spese in tempo reale e di confrontarle con il budget previsto.

1.2.4 Comunicazione

Un aspetto fondamentale dei PMIS è il supporto alla comunicazione e alla collaborazione tra i membri del team e gli stakeholder. Questi sistemi offrono piattaforme integrate per la condivisione dei documenti, la gestione delle comunicazioni e la collaborazione in tempo reale. La possibilità di centralizzare le informazioni del progetto e di renderle accessibili a tutti i partecipanti migliora la trasparenza e facilita la risoluzione dei problemi.

1.2.5 Gestione dei rischi

I PMIS sono anche strumenti cruciali per la gestione dei rischi. Permettono di identificare, analizzare e monitorare i rischi potenziali durante tutto il ciclo di vita

del progetto. I project manager possono utilizzare i PMIS per sviluppare piani di mitigazione e rispondere prontamente ai problemi emergenti, riducendo l'impatto negativo sui risultati del progetto.

1.2.6 Reportistica e analisi

Infine, i PMIS offrono capacità avanzate di reportistica e analisi. Questi sistemi possono generare report dettagliati che forniscono una visione completa delle performance del progetto, utilizzando indicatori di performance chiave (KPI) e metriche personalizzate. La possibilità di analizzare i dati storici e di progetto permette ai project manager di prendere decisioni informate e basate su evidenze, migliorando continuamente i processi di gestione.

1.3 Modelli di PMIS presenti in letteratura

La gestione delle informazioni nei progetti è un elemento cruciale per il successo delle organizzazioni, specialmente in settori complessi e dinamici. Di seguito viene fornita una panoramica dei principali modelli di Project Management Information System presentati nella letteratura scientifica, evidenziando come diverse soluzioni siano state sviluppate per affrontare le sfide specifiche di vari contesti operativi. Gli studi analizzati mostrano l'evoluzione dei PMIS e il loro impatto sulla gestione delle informazioni di progetto, migliorando l'efficienza operativa, l'integrazione dei dati e il supporto decisionale.

Un primo modello di PMIS è stato proposto da Björk (1992), questo mostra come le tecniche di modellazione concettuale orientata agli oggetti possa essere utile per la gestione delle informazioni in progetti di costruzione. L'approccio presentato mira quindi a migliorare l'efficienza e l'integrazione delle informazioni attraverso l'uso di modelli standardizzati. Tra gli obiettivi principali vi è infatti l'implementazione di standard uniformi per la rappresentazione delle informazioni per progetti operanti nel settore delle costruzioni; facilitare l'integrazione delle informazioni tra diverse fasi del ciclo di vita del progetto, dalla progettazione alla manutenzione; aumentare l'efficienza operativa riducendo errori e incoerenze nelle informazioni.

Studi come quello di von Schoultz et al. (1996) discutono l'importanza di integrare i PMIS con le applicazioni utilizzati dalle altre funzioni aziendali. Viene infatti presentato un sistema informativo integrato, utile per migliorare l'efficienza e la coordinazione nella gestione dei progetti. Il modello proposto integra tutte le informazioni rilevanti di un progetto in un'unica piattaforma, facilita la comunicazione tra i diversi membri del team di progetto e rende automatici i processi di gestione del progetto per ridurre gli errori e aumentare l'efficienza. Questi obiettivi vengono raggiunti tramite l'utilizzo di un database centralizzato che raccoglie e

gestisce tutte le informazioni del progetto: moduli per la pianificazione, la gestione delle risorse, il monitoraggio dell'avanzamento e la reportistica.

Anche [Jaafari and Manivong \(1998\)](#) presenta un nuovo approccio per la creazione di un sistema informativo di gestione dei progetti, denominato Smart Project Management Information System (SPMIS). In particolare, questo modello è pensato per affrontare la crescente complessità e incertezza dei progetti, fornendo supporto decisionale in tempo reale. Il sistema ha infatti il compito di assistere i project manager nelle decisioni, integrando informazioni e analisi intelligenti. Inoltre, è stato creato un modello flessibile che possa adattarsi ai cambiamenti e alle diverse esigenze dei progetti, automatizzando la gestione delle informazioni per migliorare l'efficienza operativa. I risultati raggiunti sono quindi una gestione centralizzata delle informazioni, migliorando la coerenza e l'accessibilità dei dati; la riduzione degli errori umani tramite l'automazione e l'aumento della produttività complessiva del progetto.

Uno strumento per la gestione delle informazioni di un singolo progetto è stato presentato da [Garcia et al. \(2016\)](#): l'articolo mostra come il plug-in MyPMP, attraverso l'utilizzo di modelli standardizzati, riesca a rappresentare e gestire le informazioni di progetto; includendo funzionalità per la pianificazione, il monitoraggio e la reportistica, consentendo quindi una gestione integrata e completa del ciclo di vita del progetto. L'utilizzo del modello proposto ha portato a una significativa riduzione dei tempi e degli sforzi necessari per la gestione delle informazioni di progetto, inoltre la standardizzazione delle informazioni ha migliorato la coerenza e l'affidabilità dei dati di progetto. In particolare, il sistema si è dimostrato flessibile e adattabile alle diverse esigenze dei piccoli progetti software.

Un sistema informativo orientato agli oggetti per la gestione dei progetti di costruzione è stato definito da [Karim and Adeli \(1999\)](#) e [Fadillah and Fitriana \(2019\)](#). Il primo è stato progettato per migliorare la pianificazione, l'ottimizzazione dei costi e la gestione delle variazioni nei progetti; il secondo sviluppa invece un modello in grado di migliorare la gestione dei dati e la creazione di report.

Sono stati anche definiti modelli più orientati al monitoraggio e controllo dei progetti, come quello presentato da [Bashashin et al. \(2016\)](#), il quale propone un sistema in grado di raccogliere, elaborare e presentare le informazioni di progetto in modo coerente e accessibile, attraverso l'utilizzo di una combinazione di tecnologie informatiche. Include anche strumenti per la gestione delle risorse, il monitoraggio dei progressi e la generazione di report, tutti integrati in un'unica piattaforma. In questo modo la soluzione formulata offre un miglioramento dell'efficienza di gestione, sia per quanto riguarda i tempi che i costi necessari per l'analisi dei dati; si ha anche un aumento della trasparenza delle operazioni, consentendo un monitoraggio più accurato e tempestivo dei progressi del progetto; inoltre, il modello offre strumenti utili per il supporto decisionale, facilitando una gestione proattiva e orientata agli obiettivi.

Con lo sviluppo sempre maggiore dei PMIS, nasce un problema legato all'aumento di complessità nella progettazione, implementazione e funzionamento dei sistemi informativi. Questo viene affrontato da [Ahlemann \(2009\)](#), il quale definisce un nuovo modello concettuale di sistema informativo di riferimento, utile per facilitare la gestione di progetti.

In ambito urbano, un sistema per la gestione simultanea di più progetti infrastrutturali nelle città in rapida espansione della Cina è stato proposto da [Li et al. \(2015\)](#). È stato utilizzato un approccio modulare che include vari componenti per la gestione dei dati di progetto, la pianificazione, il monitoraggio del progresso e la gestione delle risorse. Il design del sistema si basa su un'analisi dettagliata delle esigenze specifiche dei progetti urbani cinesi e sull'integrazione delle migliori pratiche di gestione dei progetti. Attraverso l'utilizzo di questo modello sono stati raggiunti significativi vantaggi in termini di efficienza nella gestione, riducendo i tempi e i costi associati alla raccolta e all'analisi dei dati, facilitando la coordinazione all'interno dell'organizzazione e tra i vari stakeholder dei progetti, inclusi i governi locali, i costruttori e gli urbanisti.

Per quanto riguarda il mondo della consulenza, [Teixeira et al. \(2016\)](#) definisce i processi di un PMIS necessari al miglioramento dell'efficienza aziendale e come supporto alle decisioni. In particolare, la soluzione proposta risolve il problema legato ai costi sempre maggiori per l'acquisto dei PMIS e al non supporto di tali soluzioni a esigenze specifiche di piccole e medie imprese. In questo studio i risultati principali vengono presentati tramite diagrammi UML.

Il lavoro condotto da [Waheed et al. \(2019\)](#) presenta invece un meta-modello per la gestione dell'integrazione dei progetti software, focalizzandosi sulle attività di pianificazione ed esecuzione. In questo modo si è riusciti a ridurre significativamente i tempi necessari per la pianificazione e l'esecuzione delle attività progettuali; si è inoltre migliorata l'integrazione tra le informazioni di progetto.

Un altro problema che riguarda i dati di progetto è quello legato all'utilizzo di supporti cartacei per la registrazione delle informazioni riguardando lo stato di avanzamento dei lavori e la gestione dei dati finanziari di progetto. [Fachrizal et al. \(2020\)](#) propone un sistema informativo in grado di gestire i dati di progetto, attraverso il caricamento diretto dei documenti, riducendo in questo modo la perdita di quest'ultimi. Inoltre, viene migliorata la gestione finanziaria e il controllo delle attività di progetto grazie al monitoraggio computerizzato.

L'impatto dei PMIS sui project manager e sulle performance di progetto è stato studiato da [Raymond and Bergeron \(2008\)](#). In particolare, è emerso come i sistemi informativi per la gestione dei progetti rappresentino un effettivo vantaggio per i project manager. L'uso di questi porta infatti ad un miglioramento dell'efficienza e efficacia delle decisioni manageriali, portando ad una qualità maggiore nella pianificazione, monitoraggio e controllo dei progetti. Si è osservato anche un incremento di produttività, non solo per quanto riguarda le performance individuali,

ma anche quelle di progetto.

In sintesi, la letteratura esamina una varietà di modelli di Project Management Information System, ciascuno dei quali mira a migliorare diversi aspetti della gestione delle informazioni di progetto. Dai sistemi orientati agli oggetti per i progetti di costruzione ai modelli integrati per il supporto decisionale in tempo reale, i PMIS si sono evoluti per rispondere alle crescenti esigenze di efficienza, integrazione e adattabilità. Gli studi mostrano come l'adozione di questi sistemi possa portare a significativi miglioramenti nella coerenza e affidabilità dei dati, riduzione degli errori, incremento della produttività e miglioramento della qualità della gestione dei progetti. Tuttavia, l'aumento della complessità nella progettazione e implementazione dei PMIS richiede un approccio sempre più sofisticato e modulare, capace di adattarsi ai contesti specifici delle organizzazioni.

1.4 Programmazione non lineare

In matematica, si parla di programmazione non lineare quando si ha un problema di ottimizzazione matematica in cui la funzione obiettivo e/o i vincoli sono rappresentati da espressioni non lineari. Ciò significa che almeno uno tra la funzione da ottimizzare o i vincoli è non lineare, ovvero non può essere espresso come una combinazione lineare delle variabili decisionali.

Le componenti principali di un processo di ottimizzazione non lineare sono le seguenti:

- **Funzione obiettivo:** è la funzione che si vuole massimizzare o minimizzare. In contesti non lineari, questa funzione non è rappresentabile tramite una combinazione lineare di variabili decisionali. Esempi comuni posso includere funzioni quadratiche, logaritmiche, ecc.
- **Vincoli:** sono le equazioni e/o disequazioni che le variabili decisionali devono rispettare.
- **Variabili decisionali:** le variabili su cui si basa la funzione obiettivo, che possono essere manipolate per ottimizzare la funzione.

Esistono più metodi di risoluzione per questa tipologia di problemi:

- **Metodo del gradiente:** permette di identificare i punti di massimo e minimo derivando la funzione obiettivo e i vincoli.
- **Metodi di programmazione quadratica:** utilizzati per risolvere problemi con funzione obiettivo quadratica e vincoli lineari.
- **Metodi evolutivi:** algoritmi ispirati a processi di selezione naturale (ad esempio, gli algoritmi genetici).

Le tecniche di ottimizzazione non lineare sono utilizzate in vari campi, tra cui l'ingegneria, l'economia, la finanza, la chimica, il machine learning. In ingegneria, è utilizzata per il design e l'ottimizzazione dei processi, inclusi sistemi di controllo e progettazione di strutture. In economia e finanza, questi algoritmi ottimizzano i portafogli di investimento e modellano i mercati economici complessi. In chimica, sono impiegati per ottimizzare le reazioni chimiche e i processi di produzione. Infine, nell'intelligenza artificiale, gli ottimizzatori non lineari sono fondamentali per l'addestramento di modelli di machine learning, migliorando l'accuratezza e l'efficienza degli algoritmi.

1.5 Algoritmi di ottimizzazione metaeuristica

Gli algoritmi di ottimizzazione metaeuristica rappresentano una classe di tecniche di risoluzione di problemi che mirano a trovare soluzioni approssimativamente ottimali per problemi complessi e spesso non lineari, dove metodi esatti risultano impraticabili a causa della loro elevata complessità computazionale.

Gli algoritmi in questione sono quindi delle strategie di ottimizzazione di alto livello che utilizzano approcci stocastici o probabilistici per esplorare lo spazio delle soluzioni. Sono particolarmente utili per risolvere problemi di ottimizzazione non lineare, specialmente quando il problema è di grande scala, la funzione obiettivo è discontinua o non differenziabile, si ha la presenza di molteplici minimi locali, si ha la presenza di vincoli complessi.

Tra i principali algoritmi metaeuristici ci sono:

- Algoritmi genetici (GA): ispirati al processo di selezione naturale di Darwin, utilizzano operatori genetici come selezione, crossover e mutazione per evolvere una popolazione di soluzioni candidate. Ogni generazione di soluzioni viene valutata secondo una funzione di fitness, con le migliori soluzioni che hanno maggiori probabilità di essere selezionate per la generazione successiva.
- Simulated Annealing (SA): basato sul processo di ricottura metallurgica, questo algoritmo imita il raffreddamento lento di un metallo per raggiungere uno stato di minima energia. Inizia con una soluzione iniziale e una temperatura alta, permettendo transizioni verso soluzioni peggiori per evitare minimi locali, riducendo gradualmente la temperatura e limitando le accettazioni di soluzioni peggiori man mano che il processo progredisce.
- Particle Swarm Optimization (PSO): ispirato al comportamento sociale degli stormi di uccelli e banchi di pesci, questo algoritmo modella una popolazione di particelle che si muovono attraverso lo spazio delle soluzioni, guidate sia dalla propria esperienza (memoria individuale) sia dall'esperienza dei vicini

(memoria collettiva). Le particelle aggiornano la propria posizione e velocità per convergere verso le migliori soluzioni trovate.

- Ant Colony Optimization (ACO): ispirato al comportamento delle colonie di formiche nella ricerca di cibo, utilizza un insieme di agenti (formiche artificiali) che esplorano lo spazio delle soluzioni. Le formiche depositano feromoni lungo i percorsi che attraversano, influenzando le decisioni delle formiche successive. Le soluzioni più promettenti vengono così rafforzate dal maggiore deposito di feromoni, facilitando la convergenza verso l'ottimo.

1.6 Gap Statement

L'ottimizzazione della gestione dei progetti rappresenta una sfida complessa che richiede soluzioni innovative per bilanciare tempo, costi e qualità. In questa sezione verranno presentati vari approcci presenti in letteratura per la risoluzione di problemi di schedulazione mono-progetto e multi-progetto. Attraverso l'analisi di diversi studi, vengono evidenziati i benefici e le limitazioni di questi modelli, nonché le implicazioni pratiche per i project manager che mirano a prendere decisioni più informate e strategiche.

Gli algoritmi genetici sono stati utilizzati da [Isikyildiz and Akcay \(2020\)](#) e [Pellegrino et al. \(2022\)](#) per lo sviluppo di un modello in grado di ottimizzare il tempo, il costo e la qualità nei progetti di costruzione, offrendo soluzioni ottimali che consentono ai project manager di prendere decisioni informate. La soluzione sviluppata offre infatti uno strumento per assistere i principali appaltatori nella selezione dei subappaltatori più idonei.

Una valutazione comparativa del potenziale di diversi algoritmi evolutivi nel risolvere problemi di schedulazione mono-progetto è stata condotta da [Chassiakos and Rempis \(2019\)](#). In particolare, gli algoritmi considerati mirano ad ottimizzare il trade-off tempo-costo, determinando il metodo ottimale di esecuzione delle attività in modo che il progetto sia completato al minor costo possibile e entro il termine stabilito. Non è stato però affrontato nessun aspetto riguardante le interdipendenze tra progetti, che nella realtà sono presenti in qualsiasi organizzazione.

Il lavoro condotto da [El-Abbasy et al. \(2017\)](#) presenta, invece, lo sviluppo di un modello di ottimizzazione multi-obiettivo per la schedulazione di più progetti di costruzione; con l'obiettivo di minimizzare durata totale del progetto, il costo totale, tenendo in considerazione i flussi di cassa e le fluttuazioni delle risorse, oltre alla massimizzazione del profitto. Anche [Beşikci et al. \(2015\)](#) affronta il tema della pianificazione multi-progetto, in cui l'attenzione principale è rivolta a progetti che hanno attività con modalità di utilizzo delle risorse alternative. Inoltre, non è prevista condivisione delle risorse tra i progetti durante l'orizzonte di pianificazione,

ogni risorsa è quindi assegnata ad un singolo progetto. L'obiettivo principale del modello, il quale utilizza algoritmi genetici, è quindi quello di minimizzare il ritardo totale ponderato dei progetti distribuendo in modo efficiente il budget totale tra i diversi tipi di risorse per determinare le capacità generali delle risorse e poi allocare queste risorse ai singoli progetti.

Un approccio diverso per la schedulazione multi-progetto è stato utilizzato da [Can and Ulusoy \(2014\)](#), infatti il metodo proposto si basa su una decomposizione a due stadi: nel primo viene utilizzato un algoritmo genetico per la schedulazione delle macro-attività, con l'obiettivo di massimizzare l'NPV dei flussi di cassa positivi; nel secondo, invece, le micro-attività vengono schedulate sulla base dei risultati ottenuti allo stadio precedente.

Gli algoritmi evolutivi sono stati utilizzati anche da [Elazouni and Abido \(2011\)](#) al fine di sviluppare un modello per la schedulazione dei progetti di costruzione, massimizzando i profitti dei singoli progetti all'interno di un portafoglio, permettendo ai manager di selezionare la soluzione più favorevole in base alle loro priorità. I risultati ottenuti sui dati di test sono stati ottimi, ma questi prevedevano presenza di soli due progetti all'interno del portafoglio in analisi.

Un modello per la schedulazione multi-progetto con risorse limitate, che utilizza algoritmi genetici, è stato presentato da [Gonçalves et al. \(2008\)](#). La funzione obiettivo utilizzata cerca di minimizzare una nuova misura di performance che include i ritardi e gli anticipi di un determinato progetto. L'algoritmo genetico è stato implementato e testato su vari set di problemi generati, includendo progetti con diverse dimensioni e complessità. I risultati sperimentali dimostrano che l'algoritmo proposto è efficace nel trovare soluzioni di alta qualità per il RCMPSP.

Tradizionalmente, si assume che le risorse possano essere trasferite da un progetto all'altro senza costi o tempi aggiuntivi, tuttavia questa assunzione è irrealistica. A tal proposito, [Krüger and Scholl \(2009\)](#) propongono un modello matematico per la schedulazione multi-progetto considerando i tempi di trasferimento delle risorse. Gli esperimenti computazionali sono stati condotti per testare la performance delle diverse procedure euristiche sviluppate; i risultati hanno dimostrato che le regole di priorità orientate alle risorse sono efficaci, specialmente nel contesto del multi-progetto.

Il problema della schedulazione multi-progetto è stato affrontato anche da [Singh \(2014\)](#), il quale presenta un modello che abbia come sfida principale quella di minimizzare la durata complessiva dei progetti e i costi legati alle penali per il ritardo. Inoltre, ad ogni progetto è associata una priorità; sulla base di questa viene infatti decisa l'allocazione delle risorse alle singole attività.

Per quanto riguarda i progetti di costruzione e manutenzione di impianti, [Zhu et al. \(2010\)](#) definisce un modello che ottimizzi l'efficienza e minimizzi i costi di allocazione delle risorse in un contesto multi-progetto. Questo avviene attraverso la definizione di due sotto-problemi: per primo si crea un programma fattibile per una

sequenza di attività date, in secondo luogo si cerca la sequenza valida che ottimizzi la funzione obiettivo. L'approccio proposto è stato testato su benchmark standard, mostrando notevoli miglioramenti sia in termini di efficacia che di efficienza. I risultati sperimentali indicano che l'algoritmo riesce a migliorare il valore della funzione obiettivo rispetto alle soluzioni iniziali, dimostrando la sua applicabilità nella gestione della manutenzione degli impianti.

L'analisi condotta da [Sánchez et al. \(2023\)](#) discute il problema della pianificazione di progetti multipli con risorse limitate (RCMPSP) e, in particolare, vengono analizzate diverse varianti del problema, considerando aspetti come attività, risorse, relazioni e gestione del tempo, proponendo una tassonomia per identificare e posizionare ciascuna variante. Vengono classificati e analizzati i metodi di soluzione proposti per risolvere il problema RCMPSP, evidenziando l'uso di approcci metaeuristici, algoritmi genetici e tecniche di ottimizzazione. Il testo evidenzia come questo sia un campo di ricerca attivo e in crescita, con numerose varianti e soluzioni che continuano ad evolversi.

In conclusione, gli algoritmi genetici si sono dimostrati strumenti efficaci per affrontare le complesse dinamiche della schedulazione multi-progetto. L'analisi di diversi modelli e approcci evidenzia come questi algoritmi possano non solo ottimizzare il tempo e i costi, ma anche adattarsi a scenari di progetto reali caratterizzati da interdipendenze e risorse limitate. Tuttavia, esistono ancora delle sfide, come la gestione delle risorse tra progetti multipli e l'integrazione di interdipendenze, che richiedono ulteriori ricerche e sviluppi. Dagli studi analizzati emerge infatti una lacuna nella letteratura: l'assenza di un modello per la simulazione di una generica dinamica di portfolio management.

1.7 Obiettivo

L'obiettivo di questo lavoro è di proporre un modello innovativo che supporti i manager nelle fasi preliminari della gestione multi-progetto. Questo modello mira a fornire uno strumento efficace per simulare delle metriche di progetto, adattandosi dinamicamente alle variazioni nei parametri causate da stime errate.

I lavori esistenti analizzati precedentemente si concentrano prevalentemente sulla fase di schedulazione dei progetti, trascurando l'importanza di un supporto decisionale nei casi in cui i parametri stimati, utilizzati per la programmazione iniziale, siano errati. Durante la fase iniziale di un progetto, infatti, le stime possono risultare non corrette, rendendo difficile per i project manager prendere decisioni informate. Sarebbe quindi estremamente utile disporre di uno strumento in grado di adattarsi ai cambiamenti nei parametri di input della schedulazione, come la durata delle attività, il numero di risorse disponibili e i costi associati a queste.

Attraverso il modello proposto, i project manager potranno prendere decisioni più consapevoli, migliorando l'efficienza e la riuscita complessiva dei progetti.

Capitolo 2

Approccio

Nel campo del project management, la gestione simultanea di più progetti e l'adattamento a cambiamenti imprevisti sono cruciali per garantire efficienza e ottimizzazione delle risorse. Per raggiungere gli obiettivi delineati nel capitolo precedente, è stato sviluppato un modello basato su algoritmi di ottimizzazione metaeuristica, con un focus specifico sugli algoritmi evolutivi. Gli ottimizzatori metaeuristici sono strumenti potenti che affrontano problemi di ottimizzazione complessi, in cui le soluzioni tradizionali possono risultare inefficaci, soprattutto in scenari caratterizzati da spazi di soluzione ampi e multimodali.

Il modello proposto simula la gestione di più progetti all'interno di un programma, dove le risorse sono condivise anziché assegnate a singoli progetti. Ogni programma dispone di un certo tipo e quantità di risorse, che possono essere allocate dinamicamente ai progetti che ne fanno parte. L'algoritmo evolutivo è stato invece utilizzato per decidere la priorità dei progetti all'interno di un programma, stabilendo in questo modo l'ordine migliore per la loro schedulazione, così da avere una pianificazione ottimale delle risorse e delle singole attività coinvolte. La struttura e il dettaglio del funzionamento del modello verrà presentato in questo capitolo.

2.1 Modello

In questa sezione verrà presentata la struttura utilizzata per l'implementazione del modello e delle successive simulazioni. In particolare, il sistema sviluppato può essere adoperato per ottimizzare più problemi riguardanti la gestione di progetti che utilizzano risorse condivise. Più precisamente, tra i principali problemi di ottimizzazione che il modello può risolvere vi sono:

- Fornire una schedulazione iniziale ottimale dei progetti e delle attività che ne fanno parte, rispettando i vincoli riguardanti le risorse disponibili e le

precedenze tra attività.

- Suggestire possibili variazioni nella pianificazione in caso di cambiamenti nel numero di risorse allocate ad un determinato programma.
- Suggestire possibili variazioni nella pianificazione quando le durate stimate delle singole attività vengono modificate.

2.1.1 Diagramma e notazioni utilizzate

Per la rappresentazione della struttura del modello è stato utilizzato l'UML class diagram, questo permette di avere una visione d'insieme sulle classi e le relazioni esistenti tra di esse, facilitando la comprensione del funzionamento dell'intero sistema. Ogni classe rappresenta un'entità del sistema ed è definita da un nome, degli attributi e da eventuali metodi. Le relazioni indicano come le classi interagiscono tra di loro, queste sono rappresentate tramite una linea continua ai cui estremi sono specificate le cardinalità. Se la relazione presenta un rombo su uno degli estremi, indica una composizione; ovvero che la classe "figlio" non può esistere senza la classe "padre", ovvero quella su cui è presente il simbolo.

2.1.2 Descrizione delle classi

La struttura utilizzata per lo sviluppo del modello fa riferimento a lavoro condotto da [Ottaviani et al. \(2023\)](#) in cui viene presentato un PMIS per supportare diverse funzionalità chiave del project management, tra cui la pianificazione, la programmazione e la gestione delle risorse. In Figura 2.1 viene illustrato il diagramma delle classi UML, mentre di seguito è presente una descrizione dettagliata delle classi, attributi, metodi e relazioni che lo compongono.

La classe **Organization** indica l'organizzazione che sta usando il modello per la gestione dei suoi portafogli, programmi e progetti. Ogni istanza di tale classe è identificata da un identificatore univoco (**id**). La classe **Portfolio** è usata per modellare i portafogli di un'organizzazione: un portafoglio è un insieme di programmi correlati a determinati obiettivi strategici dell'azienda. Ogni organizzazione può avere più portafogli di progetti, mentre ogni portafoglio è associato ad una sola organizzazione. I programmi che fanno parte di un determinato portafoglio sono rappresentati dalla classe **Programme**: un programma è un insieme di progetti tra di loro correlati. Un portfolio può avere più programmi, mentre un programma può far parte di un solo portfolio. Il metodo `projects_schedule()` permette l'applicazione dell'algoritmo evolutivo e, quindi, il calcolo delle priorità dei progetti e la loro conseguente ri-schedulazione. Queste ultime due classi sono entrambe identificate da una stringa univoca (**id**).

Nel modello, le risorse sono rappresentate dalla classe **Resource**. Ogni risorsa possiede un **id** univoco e ha una **designazione** e un **tipo**. Il **tipo** di risorsa può essere **work**, se la risorsa è riutilizzabile, oppure **material**, se rappresenta un materiale consumabile. Per ciascuna risorsa, vengono specificati la quantità massima (**max**) a disposizione dell'organizzazione e il costo (**cost**). Il costo è interpretato diversamente a seconda del tipo di risorsa: per le risorse di tipo **work**, rappresenta il costo giornaliero, mentre per le risorse di tipo **material**, indica il costo unitario. Ogni organizzazione può possedere più risorse, ma ciascuna risorsa è di proprietà di una sola organizzazione. La gestione e l'allocazione delle risorse avvengono a livello di programma tramite la classe di associazione **ProgrammeAllocation**. L'attributo **max** in questa classe indica la quantità di una specifica risorsa assegnata a un determinato programma. Questo approccio consente di utilizzare le risorse nei singoli progetti solo quando necessario per l'esecuzione di un'attività, evitando così l'allocazione continua di una risorsa ad un progetto per il suo intero ciclo di vita.

Più progetti, rappresentati dalla classe **Project**, possono far parte di un programma. Questa classe è identificata da un **id** e presenta tre metodi: la funzione **cost()** permette di calcolare il Budget At Completion (BAC) del progetto, considerando sia i costi diretti che indiretti; mentre **time()** ritorna la durata, in giorni, del progetto secondo la pianificazione, detta anche Planned Duration (PD); **schedule()** invece permette la schedulazione delle attività del progetto, rispettando i vincoli di precedenza e di allocazione delle risorse. Vengono inoltre specificati attraverso l'attributo **oh** i costi indiretti giornalieri del progetto; questa informazione risulta utile per il calcolo totale dei costi.

Per far sì che avvenga una pianificazione adeguata dei progetti è necessario rappresentare anche le attività, questo viene fatto attraverso la classe **Task**. Ogni attività fa parte di un solo progetto ed ha un identificatore univoco (**id**); vengono inoltre specificati i possibili costi indiretti giornalieri del task attraverso l'attributo **oh**. Le attività che fanno parte di un progetto sono caratterizzate da dei vincoli di precedenza, quindi ogni task può avere più predecessori e/o successori; ciò viene rappresentato nel modello con la relazione **precede** della classe **Task** con se stessa. Questa classe presenta inoltre due metodi: **cost()** permette di calcolare il costo totale (BAC) del task, aggregando sia i costi diretti che indiretti; la funzione **set_schedule(time)** permette di definire la schedulazione dell'attività, infatti tramite il parametro **time**, che rappresenta la data di inizio pianificata del task, vengono impostati gli attributi riguardanti la schedulazione. La classe **Schedule** rappresenta quindi la pianificazione di una determinata attività ed è caratterizzata da un **id** univoco, la data di inizio (**start**), di fine (**end**) e la durata (**duration**) stimate.

Infine, la classe **Task** è anche associata alla classe **Resource**, questa infatti indica le risorse necessarie per l'esecuzione di un'attività; in particolare, la numerosità

è specificata dall'attributo `total` della classe di associazione `Requirement`. Ogni task, per essere eseguito, può necessitare di più risorse, sia per quanto riguarda la tipologia che la numerosità.

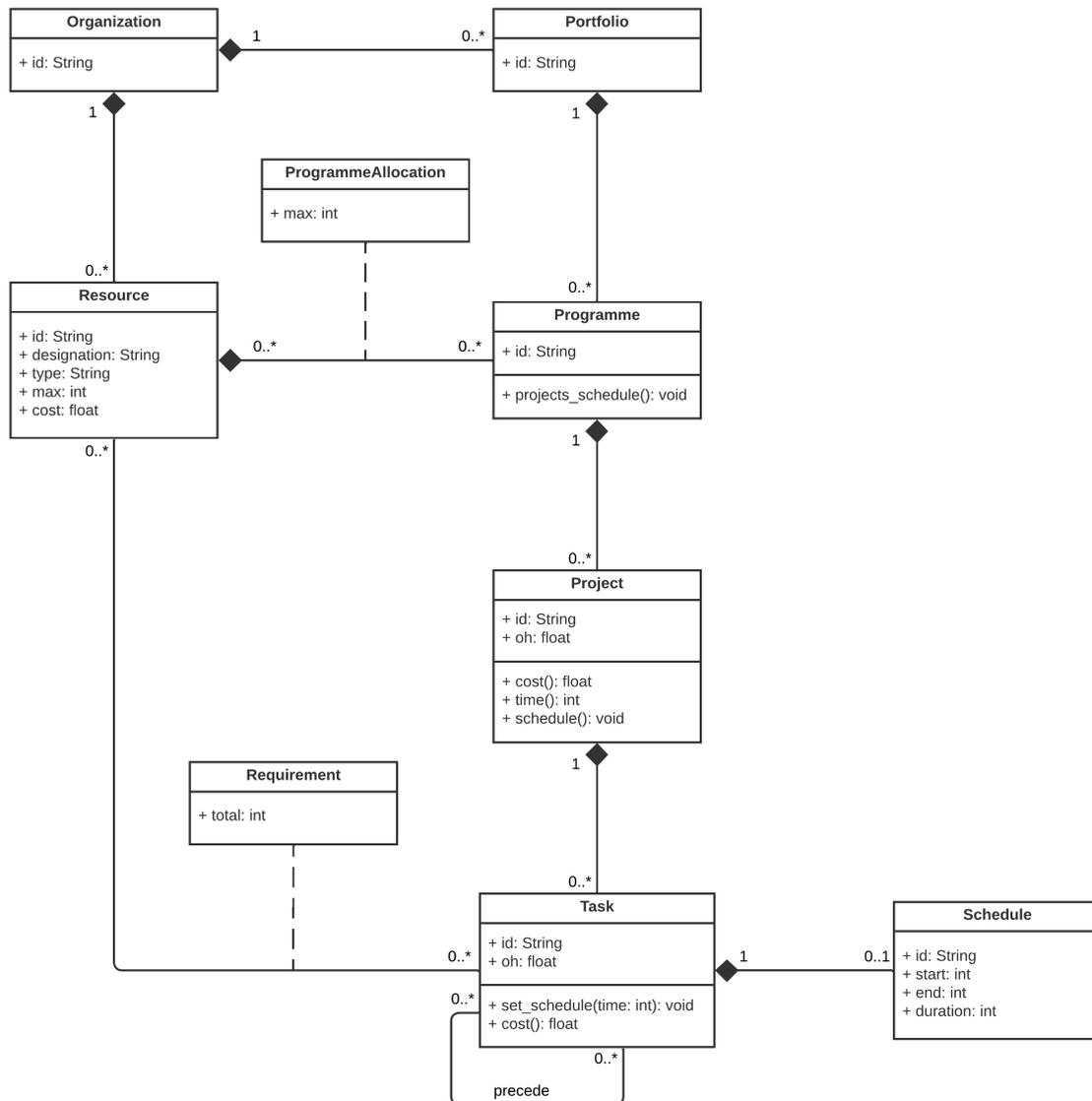


Figura 2.1: UML class diagram

2.2 Dati

Per validare il modello ed effettuare le simulazioni descritte nel capitolo successivo sono stati utilizzati dei dati riguardanti progetti con un numero di attività variabile e che utilizzano risorse condivise. Nello specifico, sono stati utilizzati due set di dati (Tabella 2.1): il primo comprende informazioni su 10 progetti, ciascuno con 92 attività, mentre il secondo include 20 progetti, ognuno con 122 attività.

Tabella 2.1: Caratteristiche dei dataset utilizzati

Dataset	N. Progetti	N. Task per progetto	N. Risorse
PRG_1	10	92	4
PRG_2	20	122	4

Entrambi i dataset sono strutturati nel seguente modo: le prime tre righe contengono le informazioni relative al programma e sono seguite da blocchi relativi ai singoli progetti. In particolare, le prime due righe definiscono il numero di progetti e il numero di risorse, nella riga successiva viene definita, per ogni risorsa, la sua disponibilità, corrispondente all'attributo `max` della classe `Resource` del diagramma descritto precedentemente. Ogni blocco presenta invece le seguenti informazioni di progetto:

- Riga 1: numero di attività del progetto, incluse quelle di inizio e fine.
- Riga 2: quali risorse utilizza il progetto (1 se il progetto utilizza quella risorsa, 0 altrimenti).
- Una riga per ogni task in cui vengono definiti: durata dell'attività, il numero di risorse necessarie per ogni tipologia di risorsa (attributo `total` della classe `Requirement`), numero di successori e relativo id.

Si precisa che ogni progetto contiene due attività fittizie, con durata nulla e senza risorse associate, che rappresentano rispettivamente l'inizio e la fine del progetto. Non sono inoltre previsti vincoli temporali per l'inizio o la conclusione dei progetti. Di seguito viene riportato a titolo di esempio, la parte iniziale del dataset PRG_1.

```

10
4
35    26    28    30

92    0
1    1    1    1

```

```

0  0  0  0  0  3  2  3  4
8  0  0  0 10  3  9 14 22
10 3  8  0  0  3  8 10 31
2  0  0  0  5  3  5 40 53
4  0  0  2  6  3  6  7 25
2  7  0  6  6  3 16 34 67
6  0  0  1  0  3 11 18 19
1  6  6  7  0  3 17 42 45
6  1  7  0  0  3 12 35 39
10 0 10  0  0  3 13 53 66
6  0  0  4  8  3 14 33 70
6  3  0  5  0  3 15 44 57
8  3  0  5  6  2 40 46
5  9  8  2  5  3 30 45 51
1  0  6 10  0  2 27 84
7  0 10  0  0  2 21 26
[...]
```

Le informazioni presenti in questi set di dati non sono risultate però sufficienti. Infatti, per raggiungere lo scopo di questo lavoro è stato necessario aggiungere dei dati, definiti in modo deterministico, necessari per il funzionamento del modello. In particolare, tali informazioni riguardano:

- I costi indiretti giornalieri di ogni progetto: sono stati definiti applicando la formula seguente, dove pos_i rappresenta la posizione in cui si trova il progetto i all'interno del database; n è invece uguale al numero totale di progetti presenti nel dataset.

$$1000 \cdot \left(1 + \frac{pos_i}{n}\right) \quad (2.1)$$

- I costi indiretti giornalieri associati ad ogni task: questi costi non influenzano il funzionamento del nostro modello per le simulazioni effettuate in seguito, infatti la durata di ogni attività non è una variabile decisionale; quindi i costi indiretti rimangono fissi e, conseguentemente, anche i costi totali di tutti i task. Per questo si è deciso di porre i costi indiretti giornalieri pari a 0 per tutte le attività.
- La tipologia e la designazione corrispondente ad ogni risorsa: tutte le risorse sono state definite di tipo 'work', mentre la designazione non influisce sul modello.
- I costi associati all'utilizzo di una risorsa: sono stati definiti applicando la seguente formula, dove pos_j indica la posizione in cui si trova la risorsa j

all'interno del database; n è invece uguale al numero totale di risorse presenti nel dataset.

$$100 \cdot \left(1 + \frac{pos_j}{n}\right) \quad (2.2)$$

2.3 Implementazione

Il modello descritto è stato implementato utilizzando il linguaggio di programmazione Python. Innanzitutto, è stata progettata l'architettura del modello, definendo le componenti principali e le loro interazioni secondo la struttura descritta in precedenza (Figura 2.1).

Per rappresentare e gestire le singole attività di un progetto e i vincoli di precedenza tra queste, è stata utilizzata la libreria NetworkX, ampiamente utilizzata per lavorare con dati strutturati a forma di grafo. Si è quindi creato, per ogni progetto, un grafo orientato con ai nodi delle istanze della classe `Task` e degli archi orientati da un'attività al suo successore. In questo modo ogni progetto è descritto da un grafo con un solo nodo di inizio e di fine, corrispondenti ai due task fittizi di inizio e fine progetto. È stata così semplificata la gestione delle attività dei progetti durante la fase di schedulazione.

2.3.1 Programmazione iniziale

Inizialmente i progetti e le relative attività sono stati schedulati seguendo un approccio deterministico basato sulle risorse, in questo modo è possibile avere un benchmark con il quale confrontare le prestazioni dell'algoritmo evolutivo quando viene applicato per risolvere il problema della programmazione iniziale dei progetti. Si è deciso assegnare ai progetti un ordine di schedulazione che segue quello definito dal dataset. Inoltre, per risolvere i problemi derivanti dalla combinazione dei diversi vincoli legati alle risorse disponibili e alle relazioni tra task, si è deciso di applicare delle regole di sequenza per la definizione di quale, tra più attività concorrenti, debba avere la priorità sull'assegnazione delle risorse disponibili. In particolare, è stata utilizzata la tecnica euristica Longest Duration First, la quale dà priorità alle attività con durata maggiore. Di seguito viene presentato nel dettaglio la procedura utilizzata per la schedulazione dei task di un progetto e i parametri utilizzati:

- T : insieme dei task che fanno parte del progetto.
- $P(t)$: insieme dei predecessori del task t .
- TS : insieme dei task da schedulare.
- $ES(t)$: early start del task t .

- $D(t)$: duration del task t .
- $R(t,i)$: variabile booleana uguale a True se sono disponibili le risorse necessarie per l'esecuzione del task t , a partire dal tempo i ; uguale a False altrimenti.

```

stop = False;
while stop == False:
    for t in T:
        if t is not scheduled:
            Add predecessors to P(t);
            if P(t) is empty:
                Add t to TS;
            else if all the tasks in P(t) are scheduled:
                Add t to TS;
    Clear P(t);
    while TS is not empty:
        max_duration = 0;
        task = null;
        time = 0;
        t_plus = 0;
        while task is null:
            for t in TS:
                i = ES(t) + t_plus;
                if D_t >= max_duration:
                    if R(t,i) == True:
                        max_duration = D(t);
                        task = t;
                        time = i;
            t_plus += 1;
        Set task's schedule with start = time;
        Update resources usage;
        Remove task from TS;
        if task has not successors:
            stop = True;

```

2.3.2 Algoritmo evolutivo

Per implementare l'algoritmo evolutivo necessario all'applicazione del modello, è stato utilizzato il metodo `differential_evolution` della libreria SciPy di Python. Questa funzione consente di applicare un algoritmo di ottimizzazione evolutiva, particolarmente utile per trovare il minimo di una funzione obiettivo complessa.

Gli algoritmi di evoluzione differenziale, introdotti da [Storn and Price \(1997\)](#), sono progettati per risolvere problemi di ottimizzazione non lineare soggetti a molteplici vincoli. Nel contesto specifico, l'obiettivo è trovare l'ordinamento ottimale per la schedulazione dei progetti. Di seguito è riportato il codice utilizzato per applicare l'algoritmo, corredato del relativo commento. La Figura 2.2 mostra invece in modo schematico il funzionamento generale dell'algoritmo.

```
def objective_function(order):
    schedule_order = sorted(range(len(order)),
                            key=lambda i: order[i],
                            reverse=True)

    obj_f = 0
    for res in resources:
        res.clear_allocation()
    for j in schedule_order:
        project = self._projects[int(j)]
        project.model()
        obj_f += project.cost()
    return obj_f

def callback(order, convergence):
    fitness_history.append(objective_function(order))

generations = 50
popsize = 15
bounds = [(0, 1) for _ in projects]
differential_evolution(objective_function, bounds,
                      strategy='best1bin', maxiter=generations,
                      popsize=popsize, callback=callback)
```

Quando l'algoritmo viene applicato si ha una prima fase di inizializzazione, in cui si genera una popolazione iniziale di soluzioni candidate (individui), ognuna rappresentata da un vettore di valori reali che descrivono una possibile soluzione al problema di ottimizzazione. La dimensione della popolazione e la dimensione del vettore di soluzioni sono dei parametri che bisogna fornire all'algoritmo; nel nostro caso la dimensione della popolazione (`popsize`) è pari a 15, mentre la dimensione del vettore di soluzioni è uguale al numero di progetti all'interno del programma. Inoltre, ogni vettore di soluzione può assumere solamente valori compresi tra 0 e 1, che rappresentano la priorità dei progetti nella schedulazione; questi vincoli sono definiti dal vettore `bounds`. Viene inoltre specificato il numero massimo di generazioni (iterazioni) che l'algoritmo eseguirà attraverso la variabile `generations`.

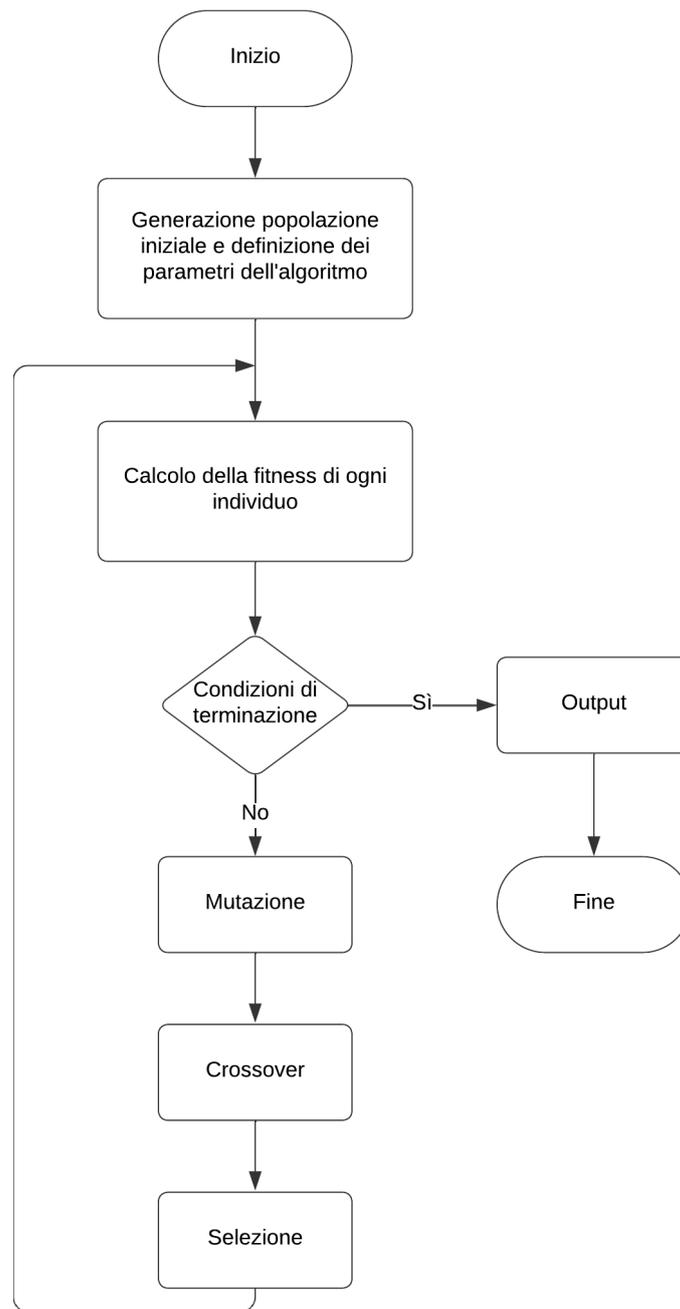


Figura 2.2: Schema a blocchi dell'algoritmo Differential Evolution

Successivamente si hanno le fasi di mutazione e crossover, per queste si è utilizzata la strategia `best1bin`, con cui nella mutazione, per ogni individuo della popolazione, si crea un vettore mutante utilizzando l'individuo migliore della

popolazione e due individui scelti casualmente. La formula di mutazione è la seguente:

$$\mathbf{v}_i = \mathbf{x}_{best} + F \cdot (\mathbf{x}_1 - \mathbf{x}_2) \quad (2.3)$$

dove:

- \mathbf{x}_{best} è l'individuo migliore della popolazione.
- \mathbf{x}_1 e \mathbf{x}_2 sono due individui scelti casualmente dalla popolazione.
- F è il fattore di scala che controlla l'intensità della mutazione.

Successivamente si utilizza il crossover binario che combina il vettore mutante \mathbf{v}_i con l'individuo corrente \mathbf{x}_i , in modo tale da creare un vettore trial \mathbf{u}_i nel seguente modo:

$$\mathbf{u}_{ij} = \begin{cases} \mathbf{v}_{ij} & \text{se } rand(0,1) \leq CR \text{ o } j = j_{rand} \\ \mathbf{x}_{ij} & \text{altrimenti} \end{cases} \quad (2.4)$$

dove:

- \mathbf{u}_{ij} è il valore della j-esima dimensione del vettore trial.
- \mathbf{v}_{ij} è il valore della j-esima dimensione del vettore mutante.
- \mathbf{x}_{ij} è il valore della j-esima dimensione dell'individuo corrente.
- $rand(0,1)$ è una funzione che genera un numero casuale tra 0 e 1.
- CR è il tasso di crossover, cioè la probabilità di prendere il valore dal vettore mutante.
- j_{rand} è un indice di dimensione casuale che garantisce che almeno una dimensione del vettore trial provenga sempre dal vettore mutante.

Infine si ha la fase di selezione in cui il vettore trial \mathbf{u}_{ij} viene confrontato con l'individuo corrente \mathbf{x}_{ij} ed in base alla funzione obiettivo utilizzata si seleziona il miglior individuo tra i due.

La funzione obiettivo (fitness) che il modello cerca di minimizzare è la somma del costo totale dei progetti. Questa è rappresentata nel codice sopra esposto dal metodo `objective_function(order)`, alla quale viene passato un vettore di numeri compresi tra 0 e 1, la cui grandezza è pari alla numerosità dei progetti, che indica l'indice di priorità di schedulazione di ogni progetto; da questo è poi infatti calcolabile l'ordine con il quale i progetti devono essere programmati, contenuto nella variabile `schedule_order`. Minimizzando il costo totale del programma, viene di conseguenza ridotta la durata programmata totale dei singoli progetti e del

programma; infatti, i costi diretti rimangono fissi, indipendentemente dall'ordine di schedulazione, mentre i costi variabili dipendono da quanto dura un progetto.

La funzione `callback(order, convergence)` è invece utile per monitorare l'andamento decrescente della funzione obiettivo nelle iterazioni, dette anche generazioni, dell'algoritmo.

Capitolo 3

Risultati

In questo capitolo viene spiegato in che modo è stato legato il genotipo dell'algoritmo al fenotipo del modello e, successivamente, verranno presentati i risultati ottenuti nelle due simulazioni effettuate sui due dataset descritti precedentemente. Entrambe le simulazioni hanno come obiettivo quello di valutare le prestazioni del modello nel risolvere problemi riguardanti la definizione della schedulazione iniziale di progetti che utilizzano risorse condivise. I risultati ottenuti saranno poi valutati utilizzando come benchmark la programmazione deterministica definita nel Capitolo [2.3.1](#).

3.1 Mappatura genotipo-fenotipo e boundaries

Nello sviluppo dell'algoritmo evolutivo, è stata implementata una codifica chiara ed efficiente per collegare il genotipo, ossia la rappresentazione codificata di una soluzione candidata nel processo di ottimizzazione, al fenotipo, che rappresenta la manifestazione concreta della soluzione determinata dal genotipo. Ogni genotipo è definito da un vettore numerico con grandezza pari alla numerosità dei progetti presenti nel programma. Questo vettore codifica l'informazione riguardante la priorità dei progetti nella schedulazione: il valore numerico presente in prima posizione indica la priorità del primo progetto, il valore in seconda posizione codifica la priorità del secondo progetto, e così via. La priorità di ogni progetto è un numero compreso tra 0 e 1, più alto è questo valore più il progetto ha priorità alta.

Questa codifica semplice e intuitiva facilita l'interpretazione dei risultati, consentendo una manipolazione efficace del genotipo durante l'evoluzione. Ogni modifica nel vettore genotipico, come risultato delle operazioni di crossover o mutazione, comporta un cambiamento immediato nella sequenza delle priorità dei progetti, influenzando così il comportamento complessivo dell'algoritmo nella ricerca della soluzione ottimale. Questa relazione diretta tra genotipo e fenotipo è

fondamentale per garantire che l'algoritmo evolutivo possa esplorare lo spazio delle possibili soluzioni in modo efficiente, adattando le priorità dei progetti in risposta alle pressioni selettive imposte dagli obiettivi dell'ottimizzazione.

3.2 Prima simulazione

Di seguito verranno presentati i risultati riguardanti l'applicazione del modello sul primo dataset (Tabella 2.1). In particolare, si cercherà di ottimizzare la schedulazione iniziale dei progetti e delle corrispondenti attività, utilizzando l'algoritmo evolutivo presentato nel Capitolo 2.3.2.

Inizialmente il programma e i corrispondenti progetti verranno schedulati secondo la procedura presentata in precedenza, verranno presentate le informazioni riguardanti e tempi e i costi derivanti da questa programmazione e in che modo è stato distribuito il carico sulle risorse. Successivamente, sarà applicato il modello e si effettuerà una comparazione con i risultati ottenuti tramite l'algoritmo evolutivo.

3.2.1 Schedulazione iniziale

Seguendo la procedura descritta nel Capitolo 2.3.1 sul primo dataset (PRG_1) comprendente 10 progetti con 92 attività ciascuno, sono stati ottenuti i risultati mostrati in Tabella 3.1. Nella prima colonna sono presenti gli identificatori dei progetti, dalla seconda in poi vengono invece mostrati l'intervallo di tempo in cui inizia il progetto e quello in cui finisce, ovvero quanto sono state schedulate le attività fittizie di inizio e fine; la durata in giorni del progetto secondo la pianificazione attuata (PD); i costi diretti, indiretti e il conseguente BAC (Budget At Completion). Si precisa che solo i costi indiretti sono influenzati dalla pianificazione dei progetti e delle attività, poiché dipendono dalla durata totale del progetto. I costi diretti, invece, rimangono costanti indipendentemente dalla pianificazione, in quanto dipendono esclusivamente dalle risorse utilizzate e dalla durata delle singole attività. L'ultima riga mostra invece le informazioni aggregate a livello di programma. Nella Figura 3.1 viene inoltre rappresentato il diagramma di Gantt dei progetti.

Le informazioni riguardanti l'utilizzo delle risorse nel tempo sono presentate nella Figura 3.2; in particolare, vengono utilizzate quattro tipologie di risorse differenti: RES1, RES2, RES3 e RES4, con una disponibilità massima, indicata nelle figure con una linea rossa tratteggiata, di 35, 26, 28 e 30 unità rispettivamente. La media, la varianza e la deviazione standard dell'utilizzo delle quattro risorse sono presenti in Tabella 3.2. Da questi dati si nota che la risorsa RES1 utilizza in media soltanto il 59,6% delle unità disponibili, mentre per RES2, RES3 e RES4 si ha una percentuale maggiore, pari rispettivamente all'82,4%, 78,3% e 72,4%.

Tabella 3.1: Risultati schedulazione iniziale - PRG_1

Progetto	Inizio	Fine	PD	Costi diretti	Costi indiretti	BAC
PRJ_1	0	112	112	967.625,00	123.200,00	1.090.825,00
PRJ_2	0	191	191	868.125,00	229.200,00	1.097.325,00
PRJ_3	0	316	316	1.407.025,00	410.800,00	1.817.825,00
PRJ_4	25	390	365	1.045.025,00	511.000,00	1.556.025,00
PRJ_5	71	529	458	1.779.000,00	687.000,00	2.466.000,00
PRJ_6	2	545	543	386.825,00	868.800,00	1.255.625,00
PRJ_7	0	625	625	926.125,00	1.062.500,00	1.988.625,00
PRJ_8	5	715	710	1.030.850,00	1.278.000,00	2.308.850,00
PRJ_9	55	779	724	845.475,00	1.375.600,00	2.221.075,00
PRJ_10	666	917	251	1.671.125,00	502.000,00	2.173.125,00
PRG_1	0	917	917	10.927.200,00	7.048.100,00	17.975.300,00

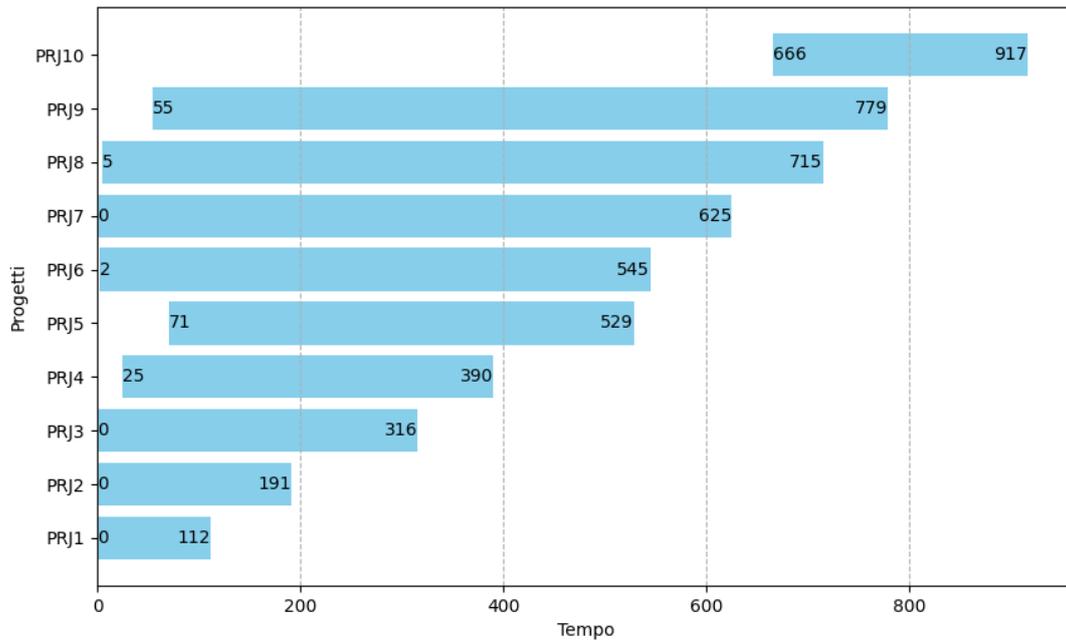


Figura 3.1: Gantt del programma PRG_1 con la schedulazione iniziale

3.2.2 Applicazione del modello

La Tabella 3.3 mostra i risultati ottenuti, in termini di tempi e costi, successivamente all'applicazione dell'algoritmo evolutivo; in Figura 3.3 è invece raffigurato il diagramma di Gantt corrispondente. È stata individuata come migliore la soluzione che schedula i progetti nel seguente ordine: PRJ_8, PRJ_9, PRJ_7, PRJ_6,

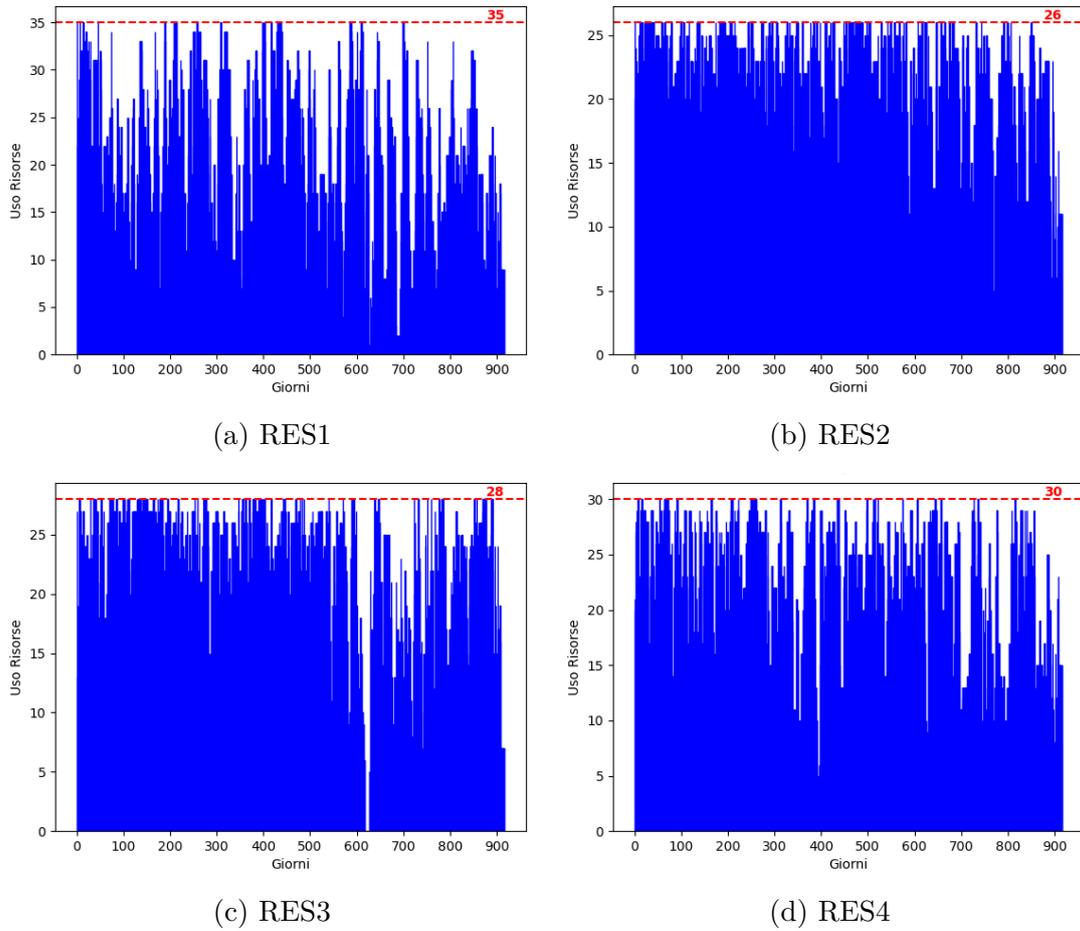


Figura 3.2: Carico delle risorse di PRG_1 con la schedulazione iniziale

Tabella 3.2: Misure del carico delle risorse di PRG_1 con la schedulazione iniziale

Risorsa	Media	Dev. Standard	Varianza
RES1	20,87	8,14	66,33
RES2	21,42	4,50	20,24
RES3	21,92	6,01	36,10
RES4	21,72	6,41	41,08

PRJ_3, PRJ_1, PRJ_2, PRJ_4, PRJ_5, PRJ_10. In questo modo si è ottenuta una riduzione del 14,4% del BAC rispetto al valore ottenuto tramite la programmazione di benchmark. Questo miglioramento è dovuto alla riduzione della durata media dei singoli progetti, e alla conseguente riduzione dei costi indiretti associati;

però, contrariamente a quanto ci si potesse aspettare, la durata totale del programma passa da 917 a 936, facendo così registrare un piccolo aumento della Planned Duration. Di seguito viene presentato l'output dell'algoritmo, il quale può dare importanti informazioni aggiuntive.

```

message: Optimization terminated successfully.
success: True
fun: 15382200.0
x: [ 4.357e-01  2.663e-01  4.999e-01  1.466e-01
      7.909e-02  8.857e-01  9.036e-01  9.881e-01
      9.130e-01  4.342e-02]
nit: 41
nfev: 6311
population: [[ 4.357e-01  2.663e-01 ... 9.130e-01  4.342e-02]
             [ 3.703e-01  3.806e-01 ... 9.573e-01  9.573e-02]
             ...
             [ 5.314e-01  7.026e-01 ... 9.111e-01  1.610e-02]
             [ 5.962e-01  4.203e-01 ... 8.969e-01  7.489e-02]]
population_energies: [ 1.538e+07  1.589e+07 ... 1.571e+07]

```

La prima riga indica che l'algoritmo è terminato con successo e che l'ottimizzazione è stata completata senza errori critici; si è quindi raggiunta una condizione di terminazione ed il processo non ha riscontrato problemi. In generale, l'algoritmo termina quando si quando i cambiamenti nelle soluzioni o nei valori della funzione obiettivo diventano molto piccoli tra generazioni successive, indicando che il processo di ottimizzazione è convergente; oppure quando si è raggiunto il numero massimo di iterazioni. Nel caso in questione il numero di iterazioni effettuate dall'algoritmo (41) è minor rispetto al massimo definito in precedenza (50); quindi si può dedurre che è stato raggiunto un punto di minimo.

Il successo dell'ottimizzazione è anche indicato dal flag booleano `success` uguale a `True`. Viene anche indicato il valore della funzione obiettivo (`fun`) per la migliore soluzione trovata e il vettore delle variabili di decisione (`x`) per cui il valore della funzione obiettivo è stato ottimizzato, rappresentante gli indici di priorità dei progetti. L'algoritmo ha completato 41 generazioni (`nit`) durante il processo di ottimizzazione, valutando la funzione obiettivo 6311 volte (`nfev`). Infine, viene specificata la popolazione finale di genotipi (soluzioni) generata e le corrispondenti funzioni obiettivo.

In Figura 3.4 è possibile visualizzare l'andamento del valore della funzione obiettivo nelle generazioni, ovvero le iterazioni dell'algoritmo. Si può notare un andamento decrescente in cui il valore minimo è raggiunto alla venticinquesima generazione.

Le informazioni riguardanti l'utilizzo delle risorse da parte delle attività dei progetti possono essere visualizzate in Figura 3.5 e in Tabella 3.4.

Tabella 3.3: Risultati del modello su PRG_1

Progetto	Inizio	Fine	PD	Costi diretti	Costi indiretti	BAC
PRJ_1	4	495	491	967.625,00	540.100,00	1.507.725,00
PRJ_2	0	555	555	868.125,00	666.000,00	1.534.125,00
PRJ_3	0	410	410	1.407.025,00	533.000,00	1.940.025,00
PRJ_4	269	645	376	1.045.025,00	526.400,00	1.571.425,00
PRJ_5	600	781	181	1.779.000,00	271.500,00	2.050.500,00
PRJ_6	0	293	293	386.825,00	468.800,00	855.625,00
PRJ_7	0	269	269	926.125,00	457.300,00	1.383.425,00
PRJ_8	0	124	124	1.030.850,00	223.200,00	1.254.050,00
PRJ_9	0	193	193	845.475,00	366.700,00	1.212.175,00
PRJ_10	735	936	201	1.671.125,00	402.000,00	2.073.125,00
PRG_1	0	936	936	10.927.200,00	4.455.000,00	15.382.200,00

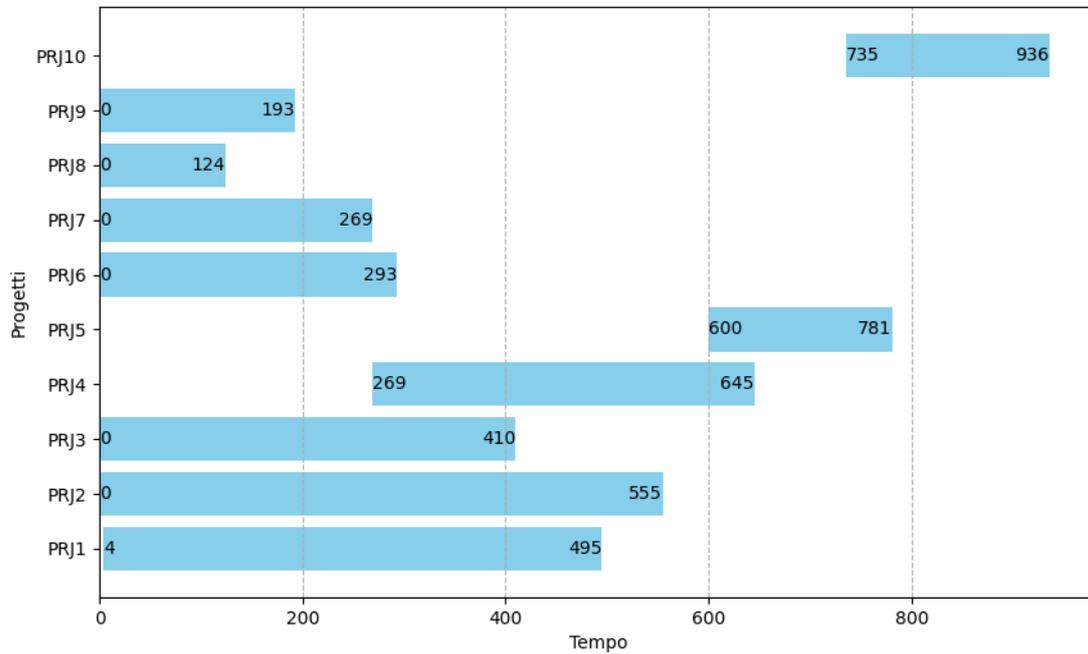


Figura 3.3: Gantt del programma PRG_1 con l'applicazione del modello

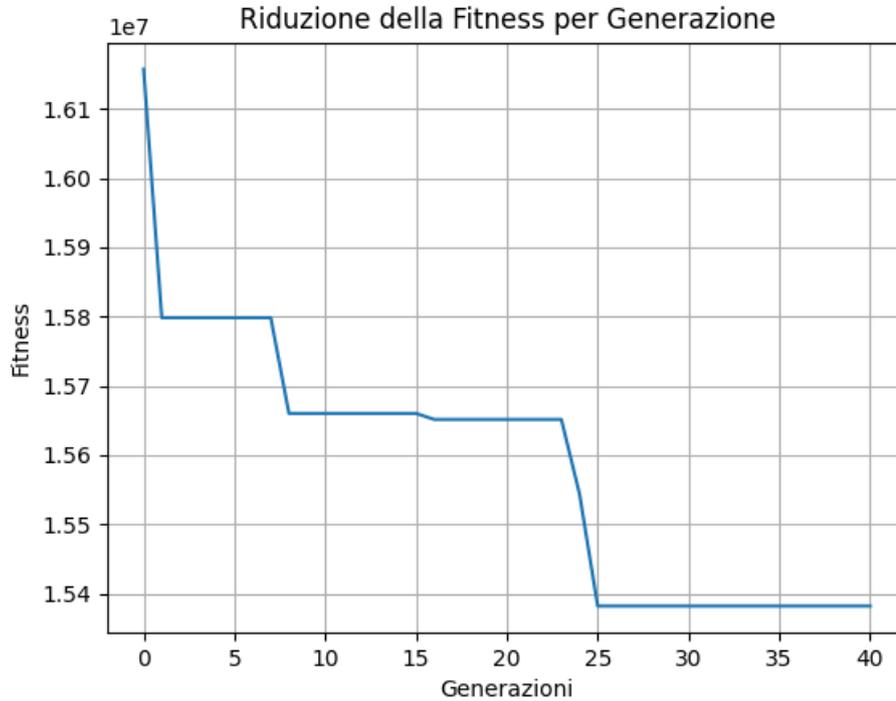


Figura 3.4: Andamento della funzione obiettivo - PRG_1

Tabella 3.4: Misure del carico delle risorse di PRG_1 con l'applicazione del modello

Risorsa	Media	Dev. Standard	Varianza
RES1	20,44	8,56	73,20
RES2	20,99	4,76	22,64
RES3	21,48	5,50	30,24
RES4	21,28	6,93	48,08

3.3 Seconda simulazione

Come fatto in precedenza, di seguito verranno presentati i risultati riguardanti l'applicazione del modello sul secondo dataset (Tabella 2.1), al fine di migliorare la schedulazione iniziale dei progetti. Anche in questo caso i progetti verranno prima programmati seguendo l'ordine del dataset e, successivamente, si utilizzerà l'algoritmo genetico per capire come cambia la schedulazione e i corrispondenti parametri.

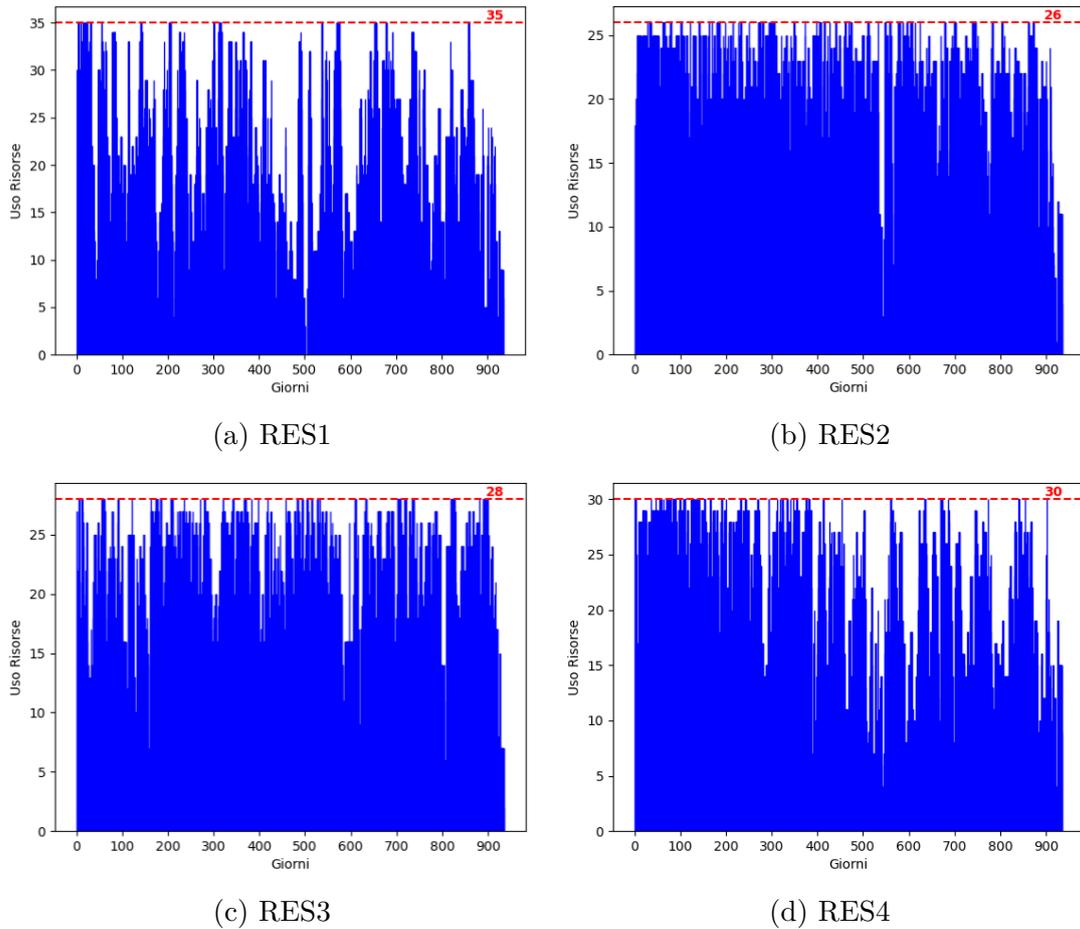


Figura 3.5: Carico delle risorse di PRG_1 con l'applicazione del modello

3.3.1 Schedulazione iniziale

Con l'applicazione del metodo descritto nel Capitolo 2.3.1 sul dataset PRG_2, sono stati ottenuti i risultati presentati in Tabella 3.5 e in Figura 3.6. Questi dati verranno utilizzati come benchmark per capire come si comporta il modello con progetti aventi caratteristiche differenti; infatti, a differenza della simulazione precedente, in questa si ha un numero maggiore di progetti (20) e anche le attività di ogni progetto hanno una numerosità più alta, pari a 122.

In Figura 3.7 è possibile visualizzare come cambia il carico delle risorse utilizzate dai progetti nel tempo. Anche in questo caso sono presenti quattro risorse con una disponibilità massima di 155 unità ognuna. In Tabella 3.6 sono inoltre presenti la media, deviazione standard e varianza del carico di ogni risorsa: si nota che RES1 e RES2 utilizzano in media il 76% circa delle unità disponibili (76,7% per RES1

e 76,3% per RES2), mentre RES3 e RES4 presentano una percentuale maggiore, pari all'80,1% per entrambe.

Tabella 3.5: Risultati schedulazione iniziale - PRG_2

Progetto	Inizio	Fine	PD	Costi diretti	Costi indiretti	BAC
PRJ_1	0	94	94	583.775,00	103.400,00	687.175,00
PRJ_2	0	85	85	545.500,00	102.000,00	647.500,00
PRJ_3	0	70	70	595.050,00	91.000,00	686.050,00
PRJ_4	0	90	90	1.778.325,00	126.000,00	1.904.325,00
PRJ_5	0	107	107	1.677.450,00	160.500,00	1.837.950,00
PRJ_6	0	158	158	1.894.375,00	252.800,00	2.147.175,00
PRJ_7	0	171	171	1.791.075,00	290.700,00	2.081.775,00
PRJ_8	0	208	208	2.450.775,00	374.400,00	2.825.175,00
PRJ_9	0	176	176	553.375,00	334.400,00	887.775,00
PRJ_10	0	225	225	1.777.125,00	450.000,00	2.227.125,00
PRJ_11	0	254	254	1.798.775,00	533.400,00	2.332.175,00
PRJ_12	0	249	249	614.600,00	547.800,00	1.162.400,00
PRJ_13	0	280	280	644.975,00	644.000,00	1.288.975,00
PRJ_14	0	269	269	584.975,00	645.600,00	1.230.575,00
PRJ_15	109	287	178	1.091.175,00	445.000,00	1.536.175,00
PRJ_16	0	303	303	1.207.025,00	787.800,00	1.994.825,00
PRJ_17	0	333	333	1.362.750,00	899.100,00	2.261.850,00
PRJ_18	229	356	127	2.466.175,00	355.600,00	2.821.775,00
PRJ_19	244	371	127	2.280.575,00	368.300,00	2.648.875,00
PRJ_20	0	420	420	2.566.900,00	1.260.000,00	3.826.900,00
PRG_2	0	420	420	28.264.750,00	8.771.800,00	37.036.550,00

Tabella 3.6: Misure del carico delle risorse di PRG_2 con la schedulazione iniziale

Risorsa	Media	Dev. Standard	Varianza
RES1	118,88	39,71	1576,76
RES2	118,31	38,42	1475,81
RES3	124,16	39,94	1594,84
RES4	124,21	40,47	1637,44

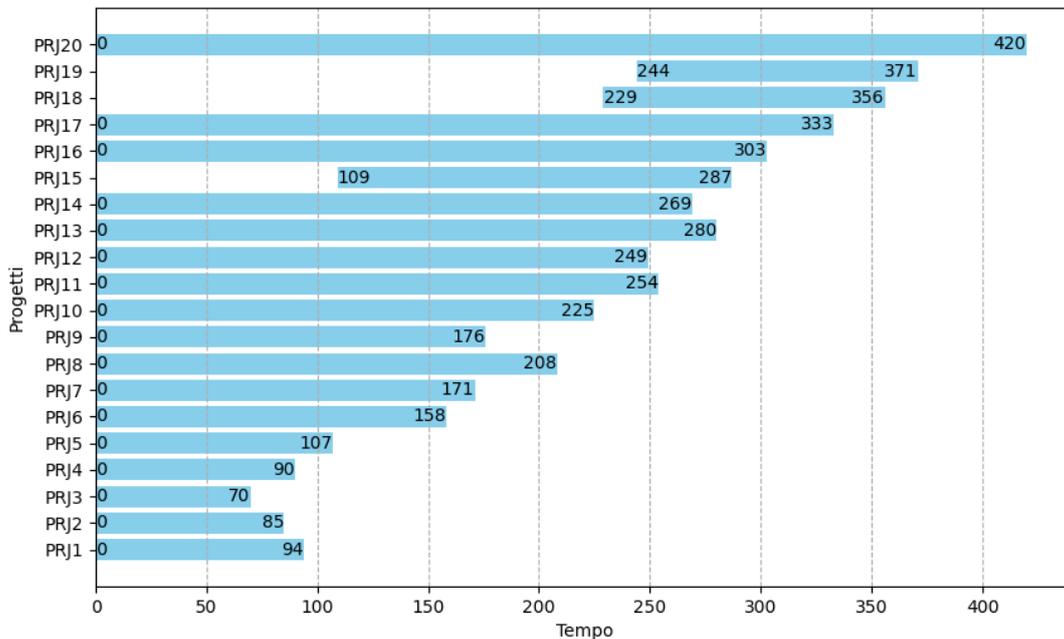


Figura 3.6: Gantt del programma PRG_2 con la schedulazione iniziale

3.3.2 Applicazione del modello

I risultati ottenuti tramite l'algoritmo evolutivo vengono presentati in Tabella 3.7, dove è possibile visualizzare le informazioni riguardanti i tempi ed i costi dei singoli progetti e del programma. In Figura 3.8 è invece rappresentato il diagramma di Gantt che mostra la collocazione temporale dei progetti programmati. Il modello ha individuato come migliore il seguente ordine di schedulazione dei progetti: PRJ_1, PRJ_11, PRJ_20, PRJ_9, PRJ_10, PRJ_13, PRJ_17, PRJ_11, PRJ_7, PRJ_14, PRJ_16, PRJ_2, PRJ_15, PRJ_18, PRJ_5, PRJ_6, PRJ_3, PRJ_8, PRJ_19, PRJ_4. I costi totali (BAC) associati alla soluzione individuata sono scesi del 6,78% rispetto al risultato di benchmark. Però, in relazione alla prima simulazione, qui i costi indiretti incidono di meno sul totale; è quindi più opportuno sottolineare che i costi indiretti, ovvero quelli su cui il modello può influire, si sono ridotti del 28,63%. La Planned Duration del programma passa invece da 420 a 413 giorni, questa rappresenta un miglioramento poco significativo poiché, come spiegato nei capitoli precedenti, l'obiettivo del modello è ridurre la durata programmata dei singoli progetti al fine di minimizzare la funzione obiettivo, non si vuole infatti necessariamente ridurre la durata dell'intero programma. Di seguito verrà presentato e commentato l'output del modello.

```
message: Optimization terminated successfully.
success: True
```

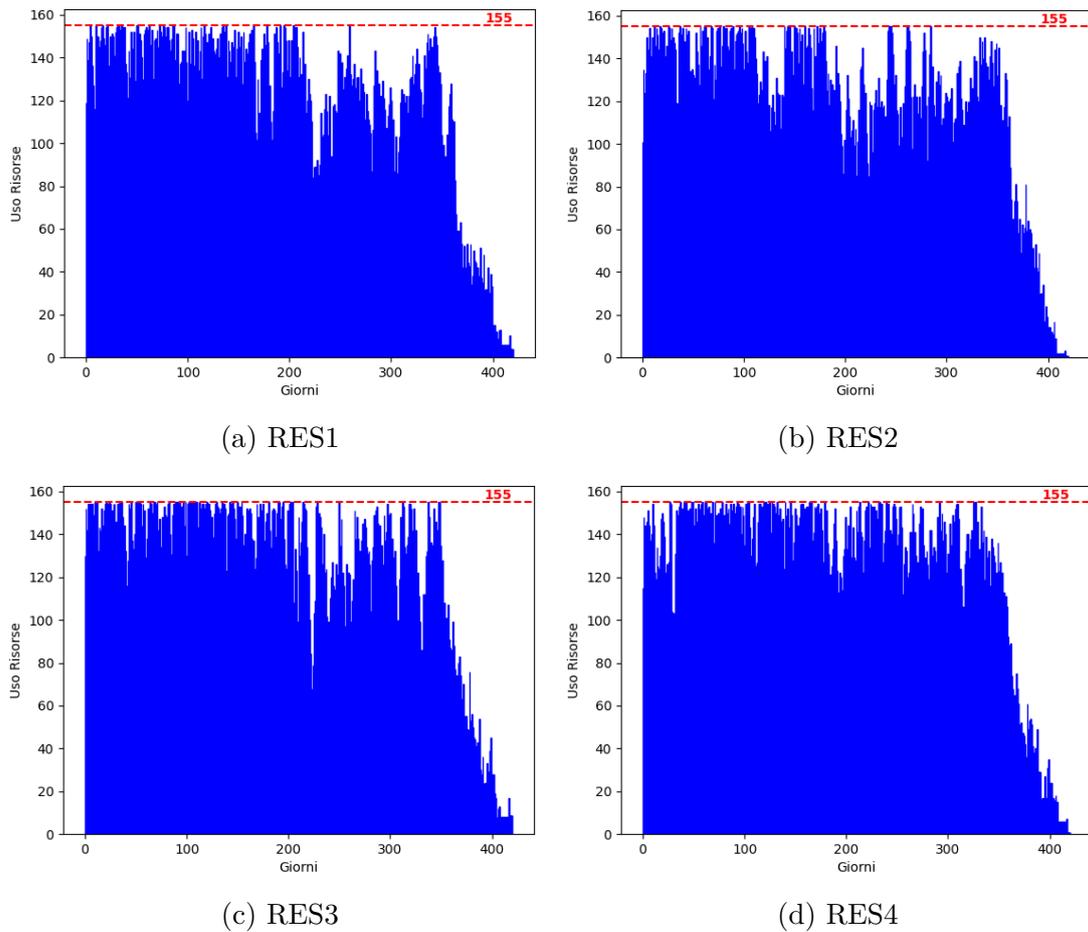


Figura 3.7: Carico delle risorse di PRG_2 con la schedulazione iniziale

```

fun: 34525450.0
x: [ 9.040e-01  4.950e-01 ...  4.929e-02  8.328e-01]
nit: 8
nfev: 2721
population: [[ 9.040e-01  4.950e-01 ...  4.929e-02  8.328e-01]
             [ 5.789e-01  3.849e-01 ...  3.968e-02  4.699e-02]
             ...
             [ 6.599e-01  8.804e-01 ...  1.410e-02  8.744e-01]
             [ 3.958e-01  4.257e-01 ...  4.486e-01  7.093e-01]]
population_energies: [ 3.453e+07  3.590e+07 ...  3.572e+07]
    
```

Come nel caso della simulazione precedente, l'algoritmo ha terminato con successo l'ottimizzazione, raggiungendo il valore della funzione obiettivo espresso da

fun. Il risultato è stato ottenuto con solo otto iterazioni (**nit**) e 2721 valutazioni della funzione obiettivo (**nfev**). La curva che rappresenta l'andamento della funzione obiettivo al progredire delle generazioni è rappresentata in Figura 3.10.

I dati riguardanti il carico delle risorse da parte delle attività dei progetti possono essere visualizzate in Figura 3.9 e in Tabella 3.8.

Tabella 3.7: Risultati del modello su PRG_2

Progetto	Inizio	Fine	PD	Costi diretti	Costi indiretti	BAC
PRJ_1	0	94	94	583.775,00	103.400,00	687.175,00
PRJ_2	0	211	211	545.500,00	253.200,00	798.700,00
PRJ_3	0	297	297	595.050,00	386.100,00	981.150,00
PRJ_4	260	413	153	1.778.325,00	214.200,00	1.992.525,00
PRJ_5	152	278	126	1.677.450,00	189.000,00	1.866.450,00
PRJ_6	168	329	161	1.894.375,00	257.600,00	2.151.975,00
PRJ_7	0	197	197	1.791.075,00	334.900,00	2.125.975,00
PRJ_8	259	373	114	2.450.775,00	205.200,00	2.655.975,00
PRJ_9	0	114	114	553.375,00	216.600,00	769.975,00
PRJ_10	0	135	135	1.777.125,00	270.000,00	2.047.125,00
PRJ_11	0	185	185	1.798.775,00	388.500,00	2.187.275,00
PRJ_12	0	103	103	614.600,00	226.600,00	841.200,00
PRJ_13	0	154	154	644.975,00	354.200,00	999.175,00
PRJ_14	0	214	214	584.975,00	513.600,00	1.098.575,00
PRJ_15	128	258	130	1.091.175,00	325.000,00	1.416.175,00
PRJ_16	0	238	238	1.207.025,00	618.800,00	1.825.825,00
PRJ_17	0	177	177	1.362.750,00	477.900,00	1.840.650,00
PRJ_18	168	290	122	2.466.175,00	341.600,00	2.807.775,00
PRJ_19	273	370	97	2.280.575,00	281.300,00	2.561.875,00
PRJ_20	0	101	101	2.566.900,00	303.000,00	2.869.900,00
PRG_2	0	413	413	28.264.750,00	6.260.700,00	34.525.450,00

Tabella 3.8: Misure del carico delle risorse di PRG_2 con l'applicazione del modello

Risorsa	Media	Dev. Standard	Varianza
RES1	122,66	37,81	1429,87
RES2	120,31	40,84	1667,87
RES3	128,11	38,81	1506,24
RES4	126,31	40,76	1661,09

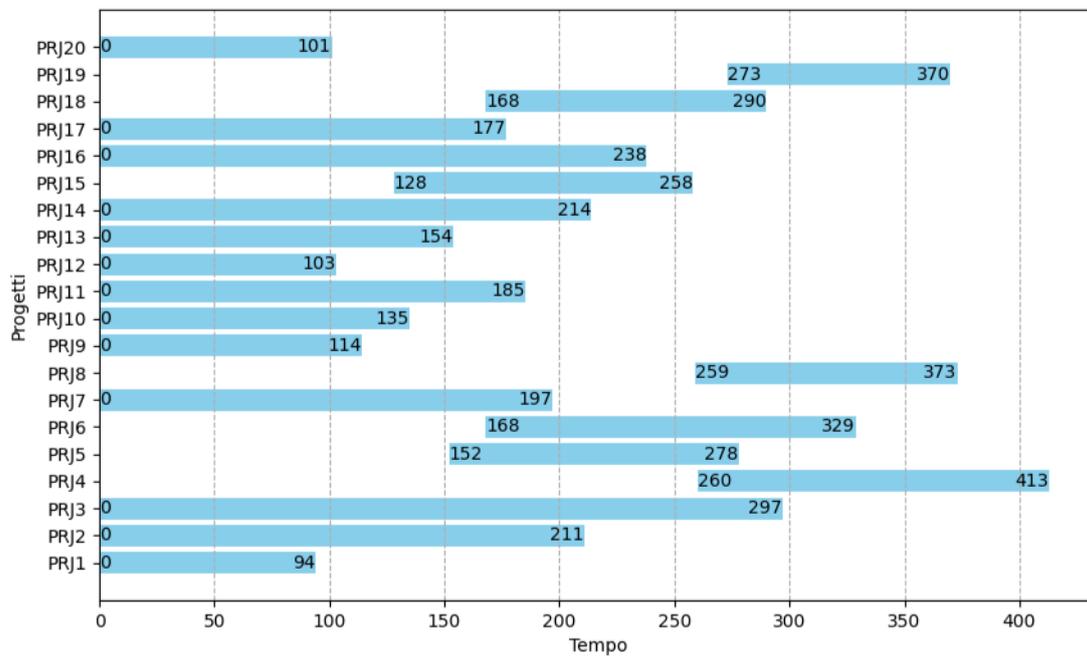
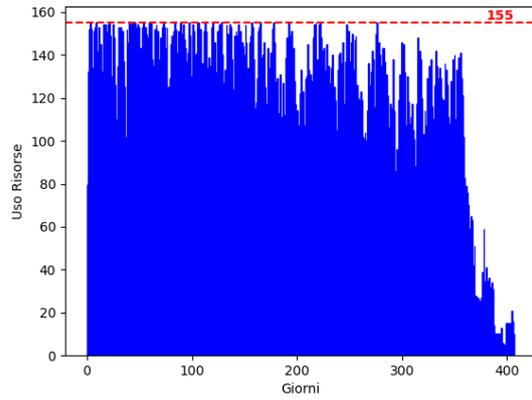
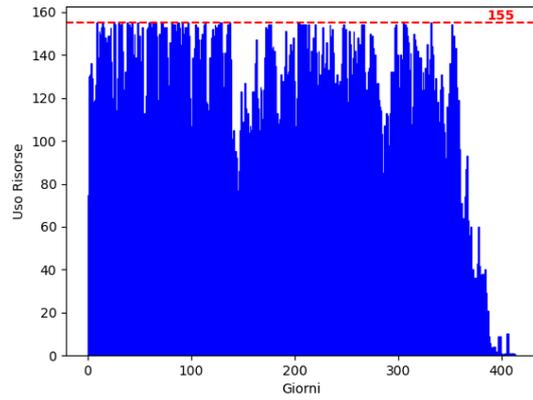


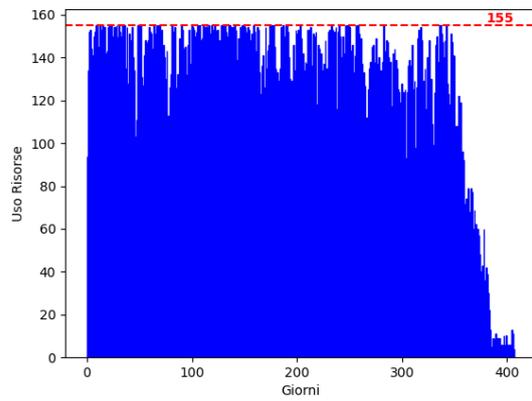
Figura 3.8: Gantt del programma PRG_2 con l'applicazione del modello



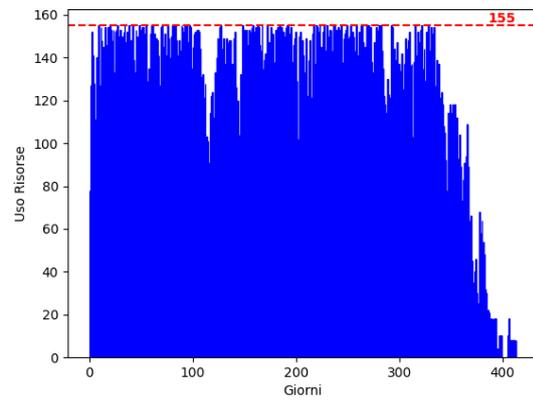
(a) RES1



(b) RES2



(c) RES3



(d) RES4

Figura 3.9: Carico delle risorse di PRG_2 con l'applicazione del modello

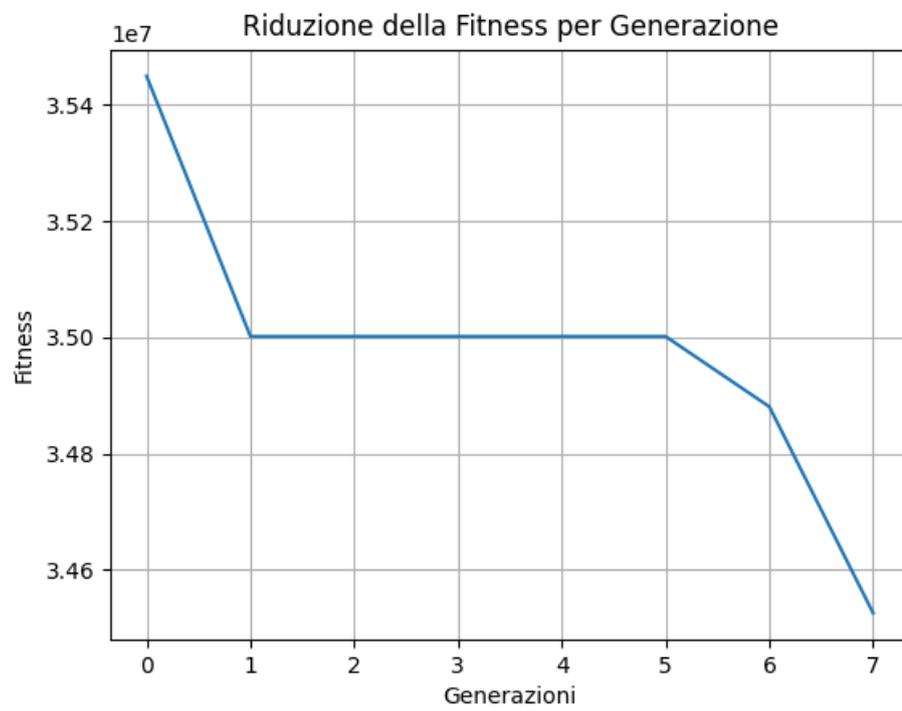


Figura 3.10: Andamento della funzione obiettivo - PRG_2

Capitolo 4

Conclusioni

Lo scopo di questo lavoro è quello di proporre un modello per il supporto alle decisioni aziendali nella gestione multi-progetto con risorse condivise. Per raggiungere questo obiettivo, è stato ideato un sistema in grado di rispondere alle variazioni nelle stime dei parametri utilizzati per la programmazione iniziale dei progetti.

Nella prima parte dello studio, sono stati introdotti i concetti di project management e di project management information system, in modo tale da dare visione generale del dominio di applicazione del modello proposto. È stata inoltre condotta una revisione della letteratura scientifica per identificare le architetture di PMIS più utilizzate nella gestione e monitoraggio dei progetti; da questa è emersa la presenza di un'alta varietà di sistemi informativi, in grado di rispondere a varie esigenze aziendali, e l'aumento della complessità nella progettazione e implementazione di questi strumenti. Successivamente, sono state brevemente presentate le principali tecniche di programmazione non lineare e gli algoritmi di ottimizzazione metaeuristica più utilizzati. È stata quindi effettuata una seconda revisione della letteratura per analizzare quelli che sono i vari approcci utilizzati per la risoluzione di problemi di schedulazione mono-progetto e multi-progetto. Si è evidenziata l'efficacia degli algoritmi di ottimizzazione metaeuristica nel risolvere questo tipo di problemi, ma anche la mancanza di un modello utile per affrontare generiche dinamiche di gestione.

Nel secondo capitolo è stato descritto l'approccio utilizzato per lo sviluppo del modello. Per prima è stata presentata la struttura del sistema attraverso il diagramma delle classi UML: sono state illustrate le entità utilizzate, la loro struttura e le relazioni che intercorrono tra di esse. In secondo luogo, si è passati alla definizione dei dati utilizzati per la validazione del modello e per l'esecuzione delle simulazioni. Infine, è stato delineato in che modo si è implementata la struttura del modello, la schedulazione iniziale utilizzata come benchmark per le simulazioni e l'algoritmo di ottimizzazione evolutiva utilizzato per trovare la soluzione ottimale.

A seguire, è stato spiegato in che modo sono stati legati genotipo e fenotipo

nel modello e sono stati presentati i risultati ottenuti dalle simulazioni effettuate su due differenti dataset. In entrambi i casi l'algoritmo è riuscito a migliorare le prestazioni in termini di costi rispetto a quelli derivanti dalla programmazione utilizzata come benchmark. Oltre alle informazioni economiche, sono state anche presentate quelle legate ai tempi e all'utilizzo delle risorse.

In accordo con il sistema informativo sviluppato da [Ottaviani et al. \(2023\)](#), il modello presentato in questo lavoro è uno strumento generico e comprensibile, utile per supportare le decisioni aziendali e simulare varie dinamiche nel contesto di project management, come la gestione della programmazione iniziale. Non si è quindi concentrati su un singolo aspetto della gestione di un portfolio di progetti, ma si è implementato un sistema in grado di adattarsi e dare supporto lungo tutto il ciclo di vita del progetto.

Detto ciò, il modello può essere migliorato attraverso delle future implementazioni, in modo tale che possa adeguarsi a varie necessità. Queste implementazioni potrebbero riguardare diverse aree critiche del project management, come ad esempio:

- La gestione dei rischi legati al progetto: ogni progetto, insieme alle attività a esso correlate, è soggetto a una serie di rischi che possono variare in base agli obiettivi prefissati e all'incertezza degli eventi futuri. I processi legati al risk management interagiscono con quelli di project management, potrebbe risultare vantaggioso includere nel modello delle funzionalità specifiche per la gestione dei rischi, consentendo così di monitorare e mitigare le potenziali minacce in modo più efficace.
- Il monitoraggio del progetto: durante la fase di esecuzione, è fondamentale implementare processi di monitoraggio e controllo accurati. Questi processi permettono di raccogliere dati sulle performance reali del progetto, consentendo di rilevare tempestivamente eventuali scostamenti rispetto alla pianificazione iniziale. Avere un sistema di monitoraggio integrato nel modello favorisce l'adozione di azioni correttive rapide ed efficaci, garantendo così un allineamento continuo agli obiettivi previsti e una gestione più reattiva del progetto.
- La qualità del progetto: oltre a rispettare i vincoli di tempo e costo, un progetto deve anche soddisfare requisiti qualitativi ben definiti. La qualità è un aspetto multidimensionale che include la conformità alle specifiche tecniche, la soddisfazione degli stakeholder e la coerenza con standard etici e professionali. Per questo motivo, è importante includere nel modello parametri qualitativi misurabili e criteri di valutazione chiari che permettano di valutare il raggiungimento degli obiettivi di qualità.

- Metodologie per la schedulazione: sviluppi futuri potrebbero riguardare sia la programmazione dei singoli progetti che quella dei programmi. Per quanto riguarda la schedulazione delle attività che fanno parte di un progetto, un fattore che potrebbe influenzare il risultato finale è l'euristica utilizzata per definire le priorità tra task concorrenti. Inoltre, altre tecniche di programmazione, oltre quella deterministica, devono essere prese in considerazione, come la programmazione probabilistica oppure l'implementazione di un algoritmo evolutivo per la schedulazione dei singoli progetti. Infine, ulteriori sforzi devono essere fatti per il miglioramento dell'algoritmo evolutivo utilizzato per la definizione delle priorità nella schedulazione di progetti che fanno parte di un programma.

Bibliografia

- Frederik Ahlemann. Towards a conceptual reference model for project management information systems. *International Journal of Project Management*, 27(1):19–30, 2009.
- MV Bashashin, DV Kekelidze, SA Kostromin, VV Korenkov, SV Kuniaev, VV Morozov, Yu K Potrebenikov, GV Trubnikov, and AV Philippov. Nica project management information system. *Physics of Particles and Nuclei Letters*, 13: 618–620, 2016.
- Umut Beşikci, Ümit Bilge, and Gündüz Ulusoy. Multi-mode resource constrained multi-project scheduling and resource portfolio problem. *European Journal of Operational Research*, 240(1):22–31, 2015.
- Bo-Christer Björk. A unified approach for modelling construction information. *Building and Environment*, 27(2):173–194, 1992.
- Anıl Can and Gündüz Ulusoy. Multi-project scheduling with two-stage decomposition. *Annals of Operations Research*, 217:95–116, 2014.
- Athanasios P Chassiakos and George Rempis. Evolutionary algorithm performance evaluation in project time-cost optimization. *Journal of Soft Computing in Civil Engineering*, 3(2):16–29, 2019.
- Mohammed S El-Abbasy, Ashraf Elazouni, and Tarek Zayed. Generic scheduling optimization model for multiple construction projects. *Journal of computing in civil engineering*, 31(4):04017003, 2017.
- Ashraf Elazouni and Mohammad Abido. Multiobjective evolutionary finance-based scheduling: Individual projects within a portfolio. *Automation in Construction*, 20(7):755–766, 2011.
- MR Fachrizal, JC Wibawa, and Z Afifah. Web-based project management information system in construction projects. In *IOP Conference Series: Materials Science and Engineering*, volume 879, page 012064. IOP Publishing, 2020.

- AP Fadillah and D Fitriana. Design of project data management information system. In *IOP Conference Series: Materials Science and Engineering*, volume 662, page 022014. IOP Publishing, 2019.
- Ivan Garcia, Carla Pacheco, Magdalena Arcilla-Cobián, and Jose Calvo-Manzano. Mypmp: A plug-in for implementing the metamodeling approach for project management in small-sized software enterprises. *Computer Science and Information Systems*, 13(3):827–847, 2016.
- José Fernando Gonçalves, Jorge JM Mendes, and Maurício GC Resende. A genetic algorithm for the resource constrained multi-project scheduling problem. *European journal of operational research*, 189(3):1171–1190, 2008.
- Setenay Isikyildiz and Cemil Akcay. Multi-objective optimization of time-cost-quality in construction projects using genetic algorithm. *Revista de la construcción*, 19(3):335–346, 2020.
- Ali Jaafari and Kitsana Manivong. Towards a smart project management information system. *International journal of project management*, 16(4):249–265, 1998.
- Asim Karim and Hojjat Adeli. Oo information model for construction project management. *Journal of Construction Engineering and Management*, 125(5):361–367, 1999.
- Doreen Krüger and Armin Scholl. A heuristic solution framework for the resource constrained (multi-) project scheduling problem with sequence-dependent transfer times. *European Journal of Operational Research*, 197(2):492–508, 2009.
- Yongkui Li, Yujie Lu, Young Hoon Kwak, and Shuang Dong. Developing a city-level multi-project management information system for chinese urbanization. *International Journal of Project Management*, 33(3):510–527, 2015.
- Filippo Maria Ottaviani, Massimo Rebuglio, Alberto De Marco, et al. Project management information system data model development and explanation. In *SIMULTECH*, pages 210–217, 2023.
- Simone Pellegrino, Massimo Rebuglio, and Giovanni Squillero. Public-private partnership: Evolutionary algorithms as a solution to information asymmetry. In *International Conference on the Applications of Evolutionary Computation (Part of EvoStar)*, pages 110–123. Springer, 2022.
- Louis Raymond and François Bergeron. Project management information systems: An empirical study of their impact on project managers and project success. *International Journal of project management*, 26(2):213–220, 2008.

- Mariam Gómez Sánchez, Eduardo Lalla-Ruiz, Alejandro Fernández Gil, Carlos Castro, and Stefan Voß. Resource-constrained multi-project scheduling problem: A survey. *European Journal of Operational Research*, 309(3):958–976, 2023.
- Amol Singh. Resource constrained multi-project scheduling with priority rules & analytic hierarchy process. *Procedia engineering*, 69:725–734, 2014.
- Rainer Storn and Kenneth Price. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11:341–359, 1997.
- Leonor Teixeira, Ana Raquel Xambre, João Figueiredo, and Helena Alvelos. Analysis and design of a project management information system: practical case in a consulting company. *Procedia Computer Science*, 100:171–178, 2016.
- Fredrick von Schoultz, Uwe Malzahn, and Ralf Schulz. *An Integrated Project Management Information System*. Citeseer, 1996.
- Fatima Waheed, Farooque Azam, Muhammad Waseem Anwar, and Ayesha Kiran. A meta-model for planning and execution activities in software project integration management. In *Proceedings of the 9th International Conference on Information Communication and Management*, pages 32–36, 2019.
- Jie Zhu, Xiaoping Li, Qi Hao, and Weiming Shen. A new approach for resource-constrained multi-project scheduling. In *Construction Research Congress 2010: Innovation for Reshaping Construction Practice*, pages 1084–1093, 2010.