



Universidade Federal de Pernambuco
Centro de Informática

Graduação em Engenharia da Computação

Escalonamento para Construção Civil

Daniele Soares Passos

Trabalho de Graduação

Recife
19 de Julho de 2016

Universidade Federal de Pernambuco
Centro de Informática

Daniele Soares Passos

Escalonamento para Construção Civil

Trabalho apresentado ao Programa de Graduação em Engenharia da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obten do grau de Bacharel em Engenharia da Computação.

Orientador: *Ricardo Martins de Abreu Silva*

Recife
19 de Julho de 2016

*Dedico este trabalho à minha mãe,
que tanto apoiou e incentivou meu
crescimento profissional.*

Agradecimentos

Agradeço a todos que fizeram parte direta ou indiretamente para a realização deste trabalho e durante toda minha graduação. Em especial, meus agradecimentos a:

Ao meu orientador, professor Ricardo Martins, pela ajuda, paciência, confiança e tempo dedicado para que este trabalho fosse realizado.

Aos meus pais, Rosemary Soares e Rogerio Passos, por todo apoio e base que me foi dada durante a graduação e em toda minha vida.

À meu namorado, Erick Lucena, por todo seu apoio e compreensão durante esse período de desenvolvimento do trabalho.

À meu primeiro colega de pesquisa, Délio Lustosa, por me ajudar nos primeiros passos que levaram à conclusão desta pesquisa.

Aos meus amigos, Maíra Cabral e Thiago Soares, por me fazerem sorrir nos momentos mais difíceis e estarem junto a mim nas horas de lazer.

À Universidade Federal de Pernambuco, especialmente ao Centro de Informática, por oferecer todo o suporte necessário para minha formação.

- *Have i gone mad?*
- *Im afraid so, but let me tell you*
something, the best people usualy are.”
—LEWIS CARROLL (Alice in Wonderland)

Resumo

Durante o processo de planejamento de um projeto na área de construção civil, as atividades que o integram devem ser previamente definidas, de forma que ocorra o máximo de aproveitamento do tempo, evitando-se a todo custo espaços ociosos. À medida que há um crescimento no número de atividades que fazem parte de um projeto, há um crescimento diretamente proporcional da complexidade envolvida no processo de ordenação do mesmo. Devido à esse fato, torna-se fundamental a utilização de ferramentas computacionais que auxiliem nesse processo. Este trabalho tem como objetivo a aplicação de algoritmos de escalonamento para assessorar no processo de alocação de recursos e planejamento de construção civil. Tratando esse problema como um análogo ao RCPSP (Resource Constrained Project Scheduling Problem), este trabalho propõe um algoritmo para encontrar uma solução ótima para o mesmo. Para alcançar esse objetivo foi desenvolvido um plugin para a ferramenta Naviswork, um software BIM (Building Information Modelling), possibilitando assim a coleta de tarefas organizadas através do software e obter como resultado o melhor escalonamento viável. Através desses processos, chegou-se a conclusão de que o algoritmo torna o planejamento de tarefas mais eficiente e produtivo.

Palavras-chave: Escalonamento, construção civil, RCPSP, BIM.

Abstract

During the planning process of a civil construction project, activities that integrate it should be previously defined in order to maximize its execution time, avoiding idle spaces at all costs. As the number of activities that are part of a project grows, there is a proportional increase in the complexity of the planning process. Due to this fact, the use of computational tools is essential to assist the whole process. This work aims to implement scheduling algorithms to assist the allocation of resources and the construction planning process. Considering this as a problem analogous to RCPSP (Resource Constrained Project Scheduling Problem), this paper proposes an algorithm to find an optimal solution for the scheduling problem. A plugin was developed for the Navisworks tool, a BIM (Building Information Modelling) software, in order to collect the tasks planned through the software. As a result, the plugin generates the best feasible schedule. Through these processes, we came to the conclusion that we get more efficient and productive tasks planning when scheduled by the proposed algorithm.

Keywords: Scheduling, civil construction, RCPSP, BIM

Sumário

1	Introdução	1
1.1	Motivação	1
1.2	Objetivo	3
1.3	Estrutura do Trabalho	4
2	Revisão da Literatura	5
2.1	Uma breve história sobre escalonamento de tarefas	5
2.2	BIM (Building Information Modelling)	7
2.2.1	Dimensões do BIM	8
2.2.2	Ferramentas BIMs	8
2.3	Modelagem do problema RCPSP	10
2.4	Algoritmo de Construção de Escalonamento	12
3	Metodologia Utilizada	15
3.1	Algoritmo de Escalonamento	15
3.2	Plugin do Navisworks	17
3.3	Caso de Estudo	19
4	Resultados e Análise	21
4.1	Configuração do Algoritmo Genético	21
4.2	Resultados dos Escalonamentos	21
4.2.1	Primeiro experimento	22
4.2.2	Segundo experimento	24
5	Conclusão	25

Lista de Figuras

2.1	Exemplo de diagrama de fluxo das atividades do projeto [JMM05].	11
2.2	Pseudocódigo básico de um algoritmo genético [MM08].	12
2.3	Algoritmo de construção do escalonamento [JMM05].	14
3.1	Fluxo completo do algoritmo de escalonamento de tarefas.	16
3.2	Fluxo completo do algoritmo de escalonamento de tarefas.	18
3.3	Construção feita no Revit.	19
4.1	Diagrama de Gantt das tarefas escalonadas no Navisworks.	22

Lista de Tabelas

2.1	Principais dimensões do BIM.	9
3.1	Tarefas planejadas à mão no Navisworks (35 tarefas).	20
4.1	Parâmetros do algoritmo genético.	21
4.2	Capacidade de recursos para o primeiro experimento.	22
4.3	Tarefas do Navisworks após o escalonamento (35 tarefas).	23
4.4	Diferenças dos tempos de execução dos projetos.	24
4.5	Resultado da função fitness para cada projeto	24

CAPÍTULO 1

Introdução

Neste capítulo será apresentado a motivação para a realização deste trabalho, os objetivos esperados e como este trabalho está estruturado.

1.1 Motivação

À medida que a complexidade dos projetos aumenta, a exigência de um planejamento organizado e escalonado é acentuada. A necessidade de um planejamento de projetos de construção mais organizado é influenciada por uma variedade de fatores, como por exemplo o número de atividades, recursos disponíveis e o tamanho do projeto.

Planejar um projeto de construção envolve a análise lógica do projeto, seus requisitos, e os planos para sua execução. Isso inclui considerar as restrições existentes e recursos disponíveis que irão influenciar no processo de execução do projeto. Esses recursos podem ser materiais, alojamentos, espaço do escritório, utilidades temporárias como um todo, e assim por diante.

A fim de planejar um projeto de construção civil, atividades devem ser definidas detalhadamente para que uma comunicação de ponto a ponto seja precisa e evite espaços ociosos e perda de tempo. A quantidade de detalhes determina o número de atividades contidas no planejamento e escalonamento do projeto. À medida que o número de atividades do projeto aumenta, também aumenta a complexidade de ordenação do projeto, então a necessidade de um projeto bem escalonado cresce [Pat04].

Um dos grandes problemas na gestão de projetos de construção civil é, de forma geral, a falta de informações necessárias para a tomada de decisões. A fim de manter o projeto funcionando, as decisões têm de ser tomadas a medida de que as informações necessárias para uma escolha ótima ainda estão sendo coletadas[AB14]. A necessidade de planejar também cresce ao passo que o número de atividades cuja execução é possível de ser feita paralelamente à outras ou quando atividades precisam ser executadas sequencialmente de acordo com uma ordem de prioridade específica. Quanto mais cresce a quantidade de tarefas e recursos, mais difícil torna-se o seu planejamento. Por esse motivo, começaram a ser utilizados softwares de planejamento para tornar mais eficiente esse processo.

O uso de computadores e softwares de planejamento de projetos se tornou e continuará sendo essencial para o desenvolvimento da indústria de construções, isso é especialmente verdadeiro para organização de atividades do projeto e gerenciamento de recursos. Esses softwares são melhores aplicados quando integrados às principais ferramentas de construção, como por exemplo os softwares CAD para arquitetura e planejamento.

Um grande problema na execução do planejamento da construção é a falta de comunicação

entre as equipes. Durante muito tempo era frequente a presença de erros graves na execução de um certo projeto, resultado da comunicação ineficaz entre os vários participantes nessa obra. Para que se diminuam esses problemas seria necessário melhorar a comunicação entre especialistas e a partilha de informação.

Como tentativa de resposta a este problema, foi criado o BIM (Building Information Modeling). Os BIMs são softwares que permitem aos seus usuários compartilharem e acrescentarem informações relevantes sobre o processo de construção uns com os outros. BIM é uma nova maneira de representação, dentro dos sistemas CAD porque permitem não só a visualização em 3D da modelagem do projeto, mas também a gestão de informação como tempo ou custo do projeto durante todo o ciclo de vida do planejamento.

Ferramentas CAD (Computer Aided Design ou desenho assistido por computador) são sistemas de computador que ajudam na criação, modificação, análise e otimização do design. São ferramentas fortemente utilizadas no planejamento de construção civil para melhorar a produtividade e qualidade do modelo de projeto [Nar08]. Softwares CAD, como o Revit e o Navisworks da empresa Autodesk [Aut12b], já incluem BIM por padrão nas suas funcionalidades. Com o BIM pode incluir tempo e custo, eles são muito usados para planejar as tarefas do projeto [Cro12]. O BIM será melhor detalhado na seção 2.1 do trabalho.

Claramente, planejamento é o elemento chave do gerenciamento do projeto de construção. Para o contratador da construção, o planejamento começa na aquisição de trabalho e continua durante o processo de construção. Ele possui um papel crucial a fim de entregar um projeto de construção dentro do prazo pré-estabelecido, no custo e na qualidade certa exigidas pelo cliente. Gerentes operacionais procuram o encarregado do planejamento para guiar o trabalho, a fim de fazer previsões dos eventos futuros: o quê e quando vai acontecer.

Identificar o caminho crítico do projeto é um dos fundamentos do planejamento. Por isso, é importante sempre identificar as atividades necessárias, o tempo que cada uma delas pode levar (considerando possíveis atrasos), ordenar essas atividades respeitando umas as outras (respeitando a precedência e prioridade de cada uma) e no final gerar um diagrama de rede lógica para ilustrar graficamente as atividades a serem executadas. A análise de rede requer construir um diagrama (ou grafo) de fluxo que representa as atividades a serem completadas e estimar a duração de cada uma delas.

Através do diagrama de fluxo é possível também escalonar as atividades a fim de otimizar o tempo e o uso de recursos, e também reduzir o tempo ocioso entre atividades. O escalonamento do projeto permite determinar a melhor forma de organizar tarefas, calculando a menor duração do projeto, o que envolve a alocação de recursos limitados para definir o início e fim da execução das atividades [MM08]. Esse processo geralmente é feito à mão ou através de algum software de gestão de tarefas, mas pode ser otimizado utilizando algoritmos específicos de alocação de recursos.

Durante os últimos anos, vários procedimentos heurísticos foram desenvolvidos para resolver o problema de escalonamento de tarefas e alocação de recursos para construção civil. Porém, esses procedimentos ainda precisam ser otimizados ou refeitos para encontrar uma solução ótima [MM08]. Esse problema é chamado de RCPS (Resource Constrained Project Scheduling Problem ou problema de escalonamento de projeto com recursos restritos) e será usado para modelar o problema de escalonamento para construção civil deste trabalho.

O RCPSP é um problema difícil de otimização combinatorial para o qual várias pesquisas extensivas já foram consagradas devido ao desenvolvimento de algoritmos eficientes que buscam resolver esse problema. Durante os últimos anos várias heurísticas têm sido desenvolvidas para esse problema. Contudo, elas frequentemente falham em achar uma solução quase ótima. O problema RCPSP pertence à classe dos problema de otimização NP-difíceis [Bla83], tornando quase que indispensável o uso de procedimentos de solução heurísticos ao tentar solucionar grandes instâncias do problema. O problema RCPSP será modelado em detalhes no Capítulo 2 do trabalho.

Para propor a solução ótima do escalonamento das tarefas, foi utilizado um algoritmo genético. Algoritmos genéticos (GAs) são algoritmos de busca baseados no mecanismo de seleção natural e genética. Essa heurística é frequentemente utilizada para gerar soluções para problemas de otimização, fazendo uso de técnicas inspiradas pela evolução natural como herança, mutação, seleção e cruzamento [Gol89].

Algoritmos genéticos precisam de um decodificador, que pode se adaptar ao contexto do problema, para avaliar o cromossomo. Este trabalho utilizou um algoritmo de decodificação proposto por Mendes et. al [JMM05] para solucionar o problema RCPSP aplicado à construção civil a fim de gerar a melhor solução em tempo e recurso. O algoritmo de decodificação gera um valor que será usado para avaliar o cromossomo.

O algoritmo irá gerar uma saída de tempos de finalização para cada tarefa que serão utilizados para construir o diagrama de espaço para uma melhor visualização do escalonamento das tarefas. Diagramas de espaço são gráficos de barra que ilustram uma figura bidimensional das atividades do projeto e o escalonamento de quando elas serão executadas [MO97]. Podemos dizer que os diagramas de espaço funcionam como um calendário de atividades.

A análise de recursos, tempo, materiais, profissionais e equipamentos disponíveis são a chave para um bom gerenciamento de projeto. Porém, essa chave não se refere necessariamente apenas para construção civil. Pode ser a representação qualquer projeto que precise ser planejado e escalonado levando em consideração os recursos disponíveis para executá-lo.

1.2 Objetivo

O objetivo dessa pesquisa é aplicar o algoritmo de escalonamento proposto para o problema RCPSP no contexto do problema de planejamento de construção civil, a fim de encontrar o resultado para solucionar questões complexas de gerenciamento de recursos no cenário de construção.

O resultado ideal será na forma de um escalonamento de tarefas no tempo com o objetivo de minimizar o tempo ocioso entre as tarefas, maximizar o aproveitamento dos recursos e reduzir o tempo total de execução paralelizando a execução das tarefas quando possível.

De início será feito uma revisão sobre o estado da arte do problema de escalonamento de tarefas para construção civil e definição do cenário de comparação. Posteriormente será feita a análise dos resultados fazendo comparações entre respostas obtidas manualmente por arquitetos e a resposta resultante do algoritmo proposto.

Construção civil é uma área que apesar de usar de recursos computacionais para seu planejamento arquitetônico (ferramentas CAD), ainda falha muito na organização de tarefas, sendo

elas feitas geralmente de forma manual. Assim, o objetivo principal deste trabalho é propor uma forma mais automática de obter o melhor escalonamento de tarefas levando em conta o tempo e seus recursos disponíveis.

1.3 Estrutura do Trabalho

Este trabalho é composto por 5 capítulos. No capítulo 2 é feita uma revisão da literatura, começando por uma breve história sobre o problema de escalonamento de tarefas para construção civil desde o início do uso de gráficos de barra até o modelo mais sofisticado do BIM. Também no mesmo capítulo será detalhada a modelagem do problema RCPSP aplicado a construção civil desde a modelagem do problema RCPSP e uma revisão literária de soluções propostas para o mesmo problema de recursos.

No capítulo 3, é mostrada a metodologia que foi utilizada para obtenção e avaliação do resultado do escalonamento das tarefas. Nesse capítulo também será melhor detalhado como os dados foram obtidos e onde serão utilizados.

Após toda a base formada, o capítulo 4 mostra o caso de teste, os métodos utilizados para análise e seus respectivos resultados. No capítulo 5 são feitas as considerações finais e possíveis trabalhos futuros.

Revisão da Literatura

Essa seção fala uma breve história do escalonamento e planejamento de tarefas e desde o uso de gráficos com barras até a técnica mais sofisticada de BIM. Também nessa seção será feita uma revisão bibliográfica dos algoritmos genéticos e de escalonamento aplicados ao problema de alocação de recursos, ilustrando também a modelagem do RCPSP.

2.1 Uma breve história sobre escalonamento de tarefas

Durante a primeira metade do século 20, o gráfico de barras era a técnica dominante para planejamento e escalonamento para projetos de todos os tamanhos [AB14]. Os gráficos de barras são excelentes para mostrar atividades e os efeitos de atraso em atividades individualmente, porém, eles claramente falham em mostrar a relação entre as atividades e o efeito que o atraso pode causar sobre elas em relação ao projeto como um todo. Portanto, era preciso um novo método de abordagem do problema a fim de considerar todos os fatores que influenciam no desenvolver do projeto.

Tendo em vista esse problema, militares americanos desenvolveram uma técnica chamada de CPA (Critical Path Analysis ou Análise do Caminho Crítico) nos anos 50 [Loc74]. O grande sucesso dessa técnica levou à sua adoção por gerentes de projetos, e durante as décadas seguintes novas técnicas mais sofisticadas começaram a ser desenvolvidas por pesquisadores de universidades e indústrias e foram sendo usadas por vários tipos de projetos. Afinal, escalonamento de tarefas pode ser aplicado a qualquer tipo de problema, até mesmo para organizar a escrita de um trabalho de graduação.

A introdução dos computadores pessoais proveu um poder computacional local de "baixo custo" para todos os escritórios e áreas de construções. Isso significou a redução do tempo exigido para a preparação dos planos e produção de horários a serem seguidos, e também significou o compartilhamento de forma mais rápida para toda a equipe de construção. O sucesso do IMB PC (introduzido em 1981), seus modelos subsequentes e outros computadores alternativos resultaram na produção de várias novas aplicações de software.

A criação de dois importantes desenvolvimentos em planejamento e escalonamento aconteceu na década de 90, foram eles: CCPM (Critical Chain Project Management ou Corrente Crítica de Gerenciamento de Projetos) e Last Planner. Os dois foram resultados de uma percepção de que a adoção de sistemas de gerenciamento de projetos existentes não poderiam garantir o sucesso do projeto, mesmo com o poder computacional barato e várias ferramentas adicionais.

CCPM foca na incerteza da organização de tarefas e identificar as atividades chaves para

formar a "corrente crítica"(critical chain) para o trabalho de construção, baseados na restrição de recursos. CCPM tem ênfase na importância de focar nas atividades críticas e nos recursos necessários para completar as tarefas.

Last Planner foi desenvolvida a partir de uma pesquisa que concluiu que mesmo com a rigidez da técnica do caminho crítico, apenas 50% das atividades de um projeto de construção típico foram finalizadas no prazo. No trabalho de Ballard and Howell [BH98] foi argumentado que a técnica CPM como uma base de planejamento de produção tinha falhas fundamentais e que a produção deveria começar apenas quando todos os recursos necessários para a finalização de uma atividade estivessem disponíveis. Ou seja, não se deve apenas considerar o que precisa ser feito mas também deve ser levado em conta o que pode ser feito. O sistema Last Planner foi introduzido em 1992 e acabou se tornando uma plataforma para Lean Construction [WR90] [WJ96]. Atualmente, o Last Planner é totalmente reconhecido como uma abordagem comprovada para a produção baseada em gerenciamento de construção.

Lean Construction (ou construção enxuta) é uma combinação de pesquisa operacional e desenvolvimento prático do design e construção, sendo uma adaptação dos princípios do modelo lean de produção para o processo de construção de ponta a ponta. Essa abordagem tenta gerenciar e melhorar o processo de construção com o custo mínimo e valor máximo considerando as necessidades do projeto (Koskela et al. 2002) [Kos02].

O surgimento e aceitação do BIM (Building Information Modeling), e a prototipação virtual aconteceram nas últimas décadas como uma base para o design, produção e manutenção de várias novas construções. Essas tecnologias junto com um foco no desenvolvimento de construções sustentáveis estão influenciando o pensamento dos clientes públicos e privados que exigem novos padrões de formas de trabalhar.

A habilidade de modelar o produto a ser construído e conectar seu conteúdo com outros sistemas foi primeiramente desenvolvido nos anos 80. Em respeito ao planejamento da construção, essa habilidade ficou conhecida como planejamento 4D (4D Planning) e basicamente se resume a conectar elementos do modelo computacional ao software de gerenciamento de projeto a fim de introduzir a dimensão do tempo (a quarta dimensão) e gerar simulações mostrando como seria o procedimento da construção durante o projeto. Similarmente, também surgiu depois a ideia da quinta dimensão (5D Planning) para analisar os custos do projeto.

O uso de modelos digitais para todos os aspectos de design de construção é atualmente conhecido como BIM (Building Information Modelling ou Modelagem da Informação de Construção). O desenvolvimento de simulações gráficas realísticas para planejar os aspectos mais amplos da construção, tanto físicos quanto operacionais, é conhecido como prototipação virtual ou construção virtual. Os benefícios dessas ferramentas e técnicas já se provam em grandes construções comerciais e na infraestrutura de trabalho. Essas ferramentas estão sendo adotadas largamente em projetos de todos os tamanhos e já estão sendo reconhecidas como parte do processo de aquisições públicas. Em 2011, o governo britânico anunciou suas intenções de que até 2016 todos os contratos públicos seriam adquiridos usando BIM [4011].

O uso de computadores e softwares de escalonamento de projetos, de preferência com o modelo BIM, se tornou e continuará sendo essencial para o desenvolvimento da indústria de construções, isso é especialmente verdadeiro para organização de atividades do projeto e gerenciamento de recursos. Esses softwares são melhores aplicados quando integrados às principais

ferramenta de construção, como por exemplo as ferramentas CAD para arquitetura e planejamento.

2.2 BIM (Building Information Modelling)

BIM, sigla para Building Information Modelling, foi apresentado como uma forma de atacar os problemas de comunicação entre os vários membros de um projeto [EL11]. Para que se diminuíssem esses problemas seria necessário melhorar a comunicação entre equipes e o compartilhamento de informação. O BIM foi a solução para melhorar a produtividade em projetos de construção.

BIM pode ser descrito como uma ferramenta que permite o armazenamento e reutilização de conhecimento de informação e domínio de todo o ciclo de vida do projeto [RV08]. Portanto, o BIM tem o principal papel de coordenar e integrar a troca de informação e conhecimento entre diferentes áreas e fases do projeto.

Esse modelo é usado em softwares de bases de dados, em formato digital, de todos os aspectos que devem ser considerados na construção de um projeto, permitindo a criação de um modelo visual 3D e facilitando a visualização do resultado final do projeto em estudo.

Ele vem substituir a representação tradicional 2D, implementando visualizações de perspectivas em 3D e informações detalhadas sobre qualquer elemento que o gerente do projeto queira representar. Por exemplo, num projeto pode-se querer indicar a utilização de um material, tempo, custo ou até resistência, algo que não é possível nos modelos CAD tradicionais.

Os BIM abriram o caminho para uma comunicação mais simplificada, completa e eficiente entre os vários especialistas envolvidos num projeto (arquitetos, engenheiros, empreiteiros ou proprietários). Com este conceito, todos envolvidos no processo de construção podem visualizar o modelo de diferentes perspectivas (tendo em conta a sua especialização), acrescentar ou modificar informações a tempo real, entre outras funcionalidades. Por exemplo, um arquiteto pode inserir a sua planta e comparar se entra em conflito com alguma estrutura dos engenheiros civil ou mecânico.

As aplicações mais vigentes dos BIM são talvez as ferramentas de concepção e design de edifícios. A modelação do edifício vai além da mera realização dos esboços em papel para formato digital, portanto, é recomendável usar o software para testar vários tipos de soluções.

Nos BIMs, a visualização espacial é completa e o processo construtivo é essencial para a modelagem. Sendo este processo automático, o utilizador apenas necessita definir o tipo de vista pretendido. Pode-se dizer que as capacidades de visualização BIM possibilitam uma melhor percepção global do modelo durante todo o ciclo de vida.

A quantificação automática e precisa é uma das primeiras e principais vantagens mencionadas quando o tema se refere aos BIMs de design. Os BIM também possuem uma ferramenta de documentação de informação que permite a captura de todos os dados no momento em que são criados, guardando-os e disponibilizando-os em qualquer nível que sejam precisos.

2.2.1 Dimensões do BIM

Desde que os BIMs começaram a fazer parte da indústria de construção, detectou-se uma crescente expansão da sua utilização a todo o ciclo de vida do planejamento. Suas dimensões foram crescendo para agregar múltiplos aspectos do projeto [Smi07] [Smi14]. As dimensões mais importantes são estão descritas na Tabela 2.1

Pela Tabela 2.1, percebe-se que os BIM possuem então uma quarta dimensão (4D), que é a dimensão mais importante para este trabalho pois trata do fator tempo. A quarta dimensão é caracterizada pela capacidade de retratar o ciclo de vida da construção, estratificando o modelo por fases de execução da construção e permitindo ainda uma visão singular da evolução da construção no decorrer do período. Portanto, é a dimensão que representa o escalonamento das atividades do projeto pelo tempo considerando recursos.

O 4D tem o intuito de visualizar corretamente o planejamento de uma obra no espaço, permitindo assim a visualização mais real do andamento da obra em um software de visualização gráfica e segundo um cronograma. É uma poderosa ferramenta de projeto que apresenta como objetivo estabelecer um estudo sobre as melhores formas de desenvolver construção da obra. Com isso, é possível verificar, antecipar possíveis problemas e cortar os custos durante a execução da obra.

2.2.2 Ferramentas BIMs

Os softwares BIM mais populares são o Autodesk Revit [Aut12b] e o Autodesk Naviswork [Aut12a]. Esses programas são ferramentas comerciais distribuídos pela Autodesk, Inc.

Autodesk Revit é uma ferramenta CAD e foi especificamente construído para Building Information Modeling (BIM) para arquitetos, engenheiros de estrutura, designers e empreiteiros. O software permite que usuários arquitetem uma construção, sua estrutura e seus componentes graficamente em 3D. Também permite tomar nota do modelo com elementos de desenho em 2D e acessar informações da construção através do banco de dados do modelo. Revit é um BIM 4D com ferramentas para planejar e acompanhar vários estágios do ciclo de vida da construção, desde o conceito passando pela construção e até mesmo uma possível demolição.

Autodesk Navisworks é um pacote de revisão do projeto que integra o BIM 4D. Navisworks é principalmente utilizado na indústria de construção para complementar o pacote do modelo gráfico 3D (normalmente feitos no Autodesk Revit ou outra ferramenta CAD de arquitetura) focando mais na gestão e simulação da obra. O Navisworks é capaz de agregar diferentes projetos de especialidade e de os associar a dimensões 4D e 5D (tempo e custo). Permite que usuários explorem e combinem o modelo 3D, navegando ao seu redor em tempo real e revisando o projeto através de suas ferramentas que incluem comentários, ponto de vista, medidas e organização de tarefas. Possui suporte para adicionar plugins que complementam funcionalidades como detecção de interferência, simulação de tempo (4D) ou até mesmo renderização. As dimensões de tempo de custo podem ser definidas no próprio programa, ou podem ser importadas através de programas dedicados à organização (calendário) do processo de construção (arquivo CSV, Microsoft Project, Primavera, entre outros). Inclui ferramentas para simular e otimizar o planejamento da obra, identificar e evitar conflitos e interferências, colaborar com as equipes de projeto, e obter uma visão geral sobre os problemas potenciais antes do início da

Tabela 2.1 Principais dimensões do BIM.

Primeira Dimensão (1D) Esboço	Fase de esboço do projeto; Também são estimados o custo, a área, o tempo, acessibilidade, viabilidade, entre outros aspectos para execução do projeto.
Segunda Dimensão (2D) Vetor	Desenvolvimento da arquitetura em 2D; Documentação; Estimativa do ciclo de vida da construção; Produção de energia; Materiais que serão usados.
Terceira Dimensão (3D) Forma	Representação do projeto em 3D, incluindo renderização e simulação da estrutura; Design detalhado; Estruturas; Especificações; Detalhes finais em geral.
Quarta Dimensão (4D) Tempo	Dimensão do tempo onde podem ser definido o escalonamento de tarefas e tempo de execução do projeto; Planejamento das atividades considerando o tempo que cada uma irá consumir; Tempo de entrega dos materiais; Simulação da execução do projeto de forma visual.
Quinta Dimensão (5D) Custo	Dimensão que considera o custo detalhado do projeto, que podem incluir: custo de contratação, de mão de obra e do ciclo de vida.
Sexta Dimensão (6D) Sustentabilidade	Considera a sustentabilidade do projeto, que leva em consideração o impacto ambiental da construção.
Sétima Dimensão (7D) Performance	Avalia a performance do projeto; Manutenção do <u>BIM</u> ; Ciclo de vida do <u>BIM</u> .

construção.

Levando em conta suas qualidades e por ser um dos softwares mais utilizados, o Autodesk Navisworks foi escolhido como ferramenta de obtenção de dados (como tarefas, durações e precedências) para a aplicação no algoritmo. Essa análise será melhor detalhada na seção de metodologia deste trabalho.

Quando comparado a sistemas CAD 2D tradicionais, BIM é uma maneira mais eficiente de lidar com a conexão de informações dentro do projeto da construção. Adotar o BIM permite implementar mudanças no processo de trabalho que podem simplificar a performance dos projetos. Não é apenas uma mudança de tecnologia, é uma necessidade de mudanças substanciais no processo de planejamento a fim de obter melhoramento de produtividade.

2.3 Modelagem do problema RCPSP

O problema RCPSP pode ser definido a seguir. Um projeto consiste de $n + 2$ tarefas, onde cada atividade deve ser processada a fim de concluir o projeto. As duas atividades em destaque são as atividades de simbolizam o início e fim do projeto e podem não influenciar na execução. O projeto sempre começa pela tarefa inicial e sempre termina na tarefa final. Seja $J = \{0, 1, \dots, n, n + 1\}$ a representação do conjunto de atividades a serem escalonadas e $K = 1, \dots, k$ o conjunto de recursos. Portanto, as atividades 0 e $n+1$ são as atividades modelos, não possuem duração e representam o início o fim do projeto. As tarefas estão relacionadas a dois tipos de restrições:

- Primeiro, restrições de prioridade, as quais forçam cada atividade j a serem escalonadas apenas depois das suas atividades predecessoras, P_j , são finalizadas.
- Depois, executar as atividades requerem recursos com limites de capacidade e portanto devem ser levados em conta.

Enquanto estão sendo processadas, as tarefas j requerem $r_{j,k}$ unidades do recurso $k \in K$ durante todo o instante de tempo da sua duração d_j . O recurso k tem uma capacidade limite de R_k a cada instante de tempo. Os parâmetros d_j , $r_{j,k}$ e R_k são positivos e determinísticos. Para as atividades de início e fim do projeto temos $d_0 = d_{n+1} = 0$ e $r_0, k = r_{n+1}, k = 0$ para todo $k \in K$. O problema consiste em achar o escalonamento das atividades levando em conta as restrições de recursos e precedências, a fim de minimizar o *makespan* (C_{max}), para completar todas as tarefas.

Seja F_j a representação do tempo de finalização da atividade j . Um escalonamento pode ser representado pelo vetor de tempos de finalização (F_1, \dots, F_{n+1}) . Seja $A(t)$ o conjunto de tarefas sendo processadas no instante t .

Para representar o fluxo de atividades, com suas precedências, será usado um diagrama de redes através de um grafo de atividades. Essa rede forma um diagrama de fluxo que mostra a lógica por trás da sequência de atividades. Assim que a lógica e as durações das atividades do diagrama estão prontas, é possível calcular o tempo total para concluir o trabalho e caminho crítico pela rede, ou seja, é possível calcular a sequência de atividades que formam o maior caminho no projeto, geralmente do início até o fim. A análise também gera informações sobre

cada atividade, indicando quanto tempo está disponível até que a atividade se torne crítica para a finalização do projeto. Adicionando requisitos de recursos a cada atividade é possível calcular o total de recursos exigidos para o projeto.

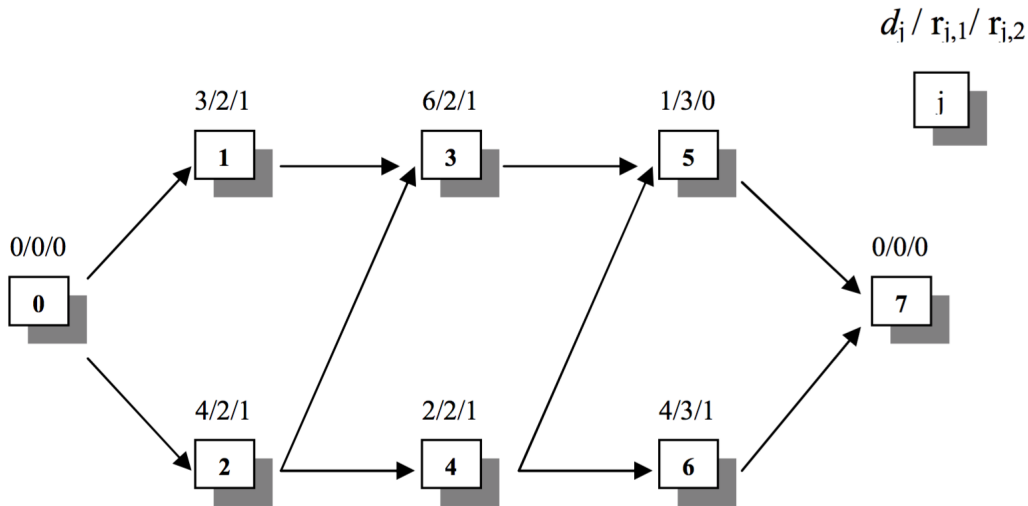


Figura 2.1 Exemplo de diagrama de fluxo das atividades do projeto [JMM05].

A Figura 2.1 representa o exemplo de diagrama de fluxo de um projeto com 6 atividades que devem ser escalonadas sujeitas a 2 recursos renováveis com a capacidade de 4 e 2 unidades respectivamente. A solução ótima do escalonamento dessas tarefas possui um *makespan* de 15 períodos de tempo.

O modelo conceitual do problema RCPSP foi descrito por Christofides et al. (1987) [CT87] da seguinte forma:

$$\text{Min } F_{n+1} \quad (2.1)$$

Sujeito a:

$$F_l \leq F_j - d_j \quad j = 1, \dots, n+1; \quad l \in P_j \quad (2.2)$$

$$\sum_{j \in A_t} r_{j,k} \leq R_k \quad k \in K; \quad t \geq 0 \quad (2.3)$$

$$F_j \geq 0 \quad j = 1, \dots, n+1; \quad l \in P_j \quad (2.4)$$

A função objetivo da equação 2.1 minimiza o tempo de finalização da atividade $n+1$, e portanto minimiza o *makespan*. A restrição da equação 2.2 impõe a precedência das relações entre as tarefas e a restrição da equação 2.3 limita a demanda de recursos, considerando a capacidade, imposta pelas atividades sendo processadas no tempo t . Finalmente, a equação 2.4 força o tempo de finalização a ser positivo.

2.4 Algoritmo de Construção de Escalonamento

Este trabalho utiliza um algoritmo genético para solucionar o problema RCPSP aplicado a construção civil a fim de gerar a melhor solução em tempo e recurso. Algoritmos genéticos (GAs) são algoritmos de busca baseados no mecanismo de seleção natural e genética. Essa heurística é frequentemente utilizada para gerar soluções para problemas de otimização, fazendo uso de técnicas inspiradas pela evolução natural como herança, mutação, seleção e cruzamento [Gol89]. A Figura 2.2 ilustra um pseudocódigo de um algoritmo genético generalizado.

procedure GENETIC-ALGORITHM

Generate initial population P_0 ;

Evaluate population P_0 ;

Initialize generation counter $g \leftarrow 0$;

While stopping criteria not satisfied repeat

Select some elements from P_g to copy into P_{g+1} ;

Crossover some elements of P_g and put into P_{g+1} ;

Mutate some elements of P_g and put into P_{g+1} ;

Evaluate some elements of P_g and put into P_{g+1} ;

Increment generation counter: $g \leftarrow g+1$;

End while

End GENETIC-ALGORITHM;

Figura 2.2 Pseudocódigo básico de um algoritmo genético [MM08].

Primeiramente, uma população inicial de possíveis soluções (individualmente diferentes) é gerada. A função de seleção é baseada na função de fitness do problema e permite escolher possíveis candidatos a reprodução. A população gerada é ordenada em ordem decrescente, do melhor para o pior. Os melhores indivíduos são copiados para a próxima população. A reprodução consiste na recombinação dos indivíduos pela operação de crossover (cruzamento). Após o crossover pode haver uma possível mutação que consiste da mudança no gene de algum indivíduo. Assim, uma nova geração é obtida a partir da população inicial. E assim por diante até atingir o critério de parada. O critério de parada é normalmente baseado no número de gerações que se quer obter.

Essas características simplificam e padronizam a meta-heurística com um conjunto de tarefas bem contidas onde apenas uma delas é dependente do contexto: a decodificação do cromossomo. Ou seja, o algoritmo pode ser adaptado a qualquer problema, variando apenas a função de decodificação e o fitness do contexto.

Para o contexto de construção civil e escalonamento de tarefas foi utilizado um algoritmo de construção do escalonamento proposto por Mendes et al. [JMM05] como decodificação para o cromossomo gerado pelo algoritmo genético.

A função proposta para decodificar as prioridades e construir os escalonamentos parametrizados é baseado em um esquema de geração de escalonamentos que incrementa com o tempo. Para cada interação g , existe um tempo de escalonamento t_g . O conjunto ativo $A_g = \{j \in J | F_j - d_j \leq t_g < F_j\}$ contém o conjunto de todas as atividades que estão ativas no tempo t_g . A capacidade do recurso k em um instante de tempo t_g é dado por:

$$RD_g(t_g) = R_k(t_g) - \sum_{j \in A_g} r_{j,k} \quad (2.5)$$

Em um projeto com n atividades, a representação do cromossomo gerado pelo algoritmo genético é dada por:

$$Cromossomo = \{gene_1, gene_2, \dots, gene_n, gene_{n+1}, \dots, gene_{2n}\} \quad (2.6)$$

Os primeiros n genes representadas em (2) serão usados para a decodificação das prioridades e os genes restantes serão usados para o calculo do atraso.

A decodificação da prioridade de cada atividade obedece a seguinte equação:

$$PRIORITY_j = \frac{LLP_j}{LCP} \times \left[\frac{1 + gene_j}{2} \right], \quad (2.7)$$

onde LLP_j é o caminho mais longo desde o início da atividade j até o final do projeto e LCP é o tamanho do caminho crítico, ou seja o caminho mais longo desde o início. A decodificação do atraso de cada atividade é dada por:

$$Delay_g = gene_{n+g} \times 1.5 \times MaxDur, \quad (2.8)$$

onde $MaxDur$ é a máxima duração de todas as atividades.

S_g contem todas as atividades que já foram escalonadas na interação g , e FS_g contem os tempos de finalização das atividades em S_g . Seja $Delay_g$ o tempo de atraso associado a interação g , e seja E_g o vetor de atividades que podem ser executadas considerando apenas a precedência no intervalo $[t_g, t_g + Delay_g]$. Ou seja:

$$E_g(t_g) = \{j \in (J \setminus S_{g-1}) | F_i \leq t_g + Delay_g (i \in P_j)\} \quad (2.9)$$

O *makespan* do escalonamento encontrado ao final da função de geração é dado pela equação:

$$makespan_n = F_{n+1} + \frac{\sum_{d=1}^L \sum_{i | dist(i)=d} F_i}{\sum_{d=1}^L \sum_{i | dist(i)=d} F_{n+1}}. \quad (2.10)$$

Esse *makespan* gerado pela decodificação do cromossomo será utilizado para avaliação e seleção da população, com o objetivo de escolher o cromossomo com melhor fitness. No contexto, a função fitness será usada para escolher o cromossomo que decodifica o escalonamento com melhor otimização de tempo e recurso.

A Figura 2.3 representa o pseudocódigo do algoritmo de construção do escalonamento para uma entrada de cromossomo. Assim, esse algoritmo será executado para cada cromossomo gerado pelo algoritmo genético.

Initialization: $g=1, t_1=0, A_0=\{0\}, \Gamma_0=\{0\}, S_0=\{0\}, RD_k(0)=R_k \ (k \in K)$

while $|S_g| < n+2$ **repeat**

{

 Update E_g

while $E_g \neq \{\}$ **repeat**

 {

 Select activity with highest priority

$j^* = \underset{j \in E_g}{\operatorname{argmax}} \{ \text{PRIORITY}_j \}$

 Calculate earliest finish time (in terms of precedence only)

$EF_{j^*} = \max_{i \in P_j} \{ F_i \} + d_{j^*}$

 Calculate the earliest finish time (in terms of precedence and capacity)

$F_{j^*} = \min \left\{ t \in [FMC_{j^*} - d_{j^*}, \infty] \cap \Gamma_g \mid r_{j^*,k} \leq RD_k(\tau), \right.$

$\left. k \in K \mid r_{j^*,k} > 0, \tau \in [t, t + d_{j^*}] \right\} + d_{j^*}$

 Update $S_g = S_{g-1} \cup \{j^*\}$, $\Gamma_g = \Gamma_{g-1} \cup \{F_{j^*}\}$

 Iteration increment: $g = g+1$

 Update $A_g, E_g, RD_k(t) \mid t \in [F_{j^*} - d_{j^*}, F_{j^*}], k \in K \mid r_{j^*,k} > 0$

 }

 Determine the time associated with activity g

$t_g = \min \{ t \in \Gamma_{g-1} \mid t > t_{g-1} \}$

}

Figura 2.3 Algoritmo de construção do escalonamento [JMM05].

Metodologia Utilizada

Nesta seção é feita a conexão entre os dois algoritmos (o genético e o de construção de escalonamento) introduzidos anteriormente, logo em seguida será explicado como o algoritmo final será usado no caso de estudo. Também é detalhado de que forma as tarefas do caso de estudo serão obtidas e finalmente o caso de estudo utilizado para os experimentos é detalhado.

3.1 Algoritmo de Escalonamento

O algoritmo genético baseado em chaves aleatórias que foi introduzido na seção 2.5 possui um decodificador que gera o *makespan* para análise da melhor resposta. Esse decodificador pode ser generalizado para utilização em qualquer contexto.

Este trabalho usa o algoritmo de construção de escalonamento de recursos introduzido na seção 2.6 como decodificador para o algoritmo genético. Sendo assim, cada cromossomo gerado aleatoriamente pelo algoritmo genético é decodificado pela função de construção de escalonamento para gerar seu *makespan* que será usado para escolher os melhores indivíduos da população.

A Figura 3.1 ilustra o diagrama de fluxo completo do algoritmo genético, levando em conta a decodificação do cromossomo.

O diagrama de fluxo segue o seguinte roteiro:

1. Inicialmente, uma população inicial de indivíduos (cromossomos) é gerada aleatoriamente. Também é nessa etapa que o grafo de tarefas, com suas durações, será inicializado. Esse grafo é usado para construir o escalonamento de todos os cromossomos.
2. Os cromossomos da população são enviados para o decodificador.
3. Cada cromossomo é então decodificado para extrair suas prioridades e atrasos.
4. Cada cromossomo gera um escalonamento diferente através do algoritmo Construct-Schedule (construtor de escalonamento detalhado no capítulo anterior), gerando um vetor de *makespan* onde cada posição representa o *makespan* de um cromossomo.
5. Então, uma validação para saber se o critério de parada foi alcançado é feita. A condição usada no critério de parada para este trabalho foi apenas a quantidade de gerações. Se o critério foi atingido pare a execução, se não continue os próximos passos.
6. Os *makespans* dos cromossomos são usados para classificar cada indivíduo da geração como elite ou não elite, ou seja, é feita uma classificação dos cromossomos que tem

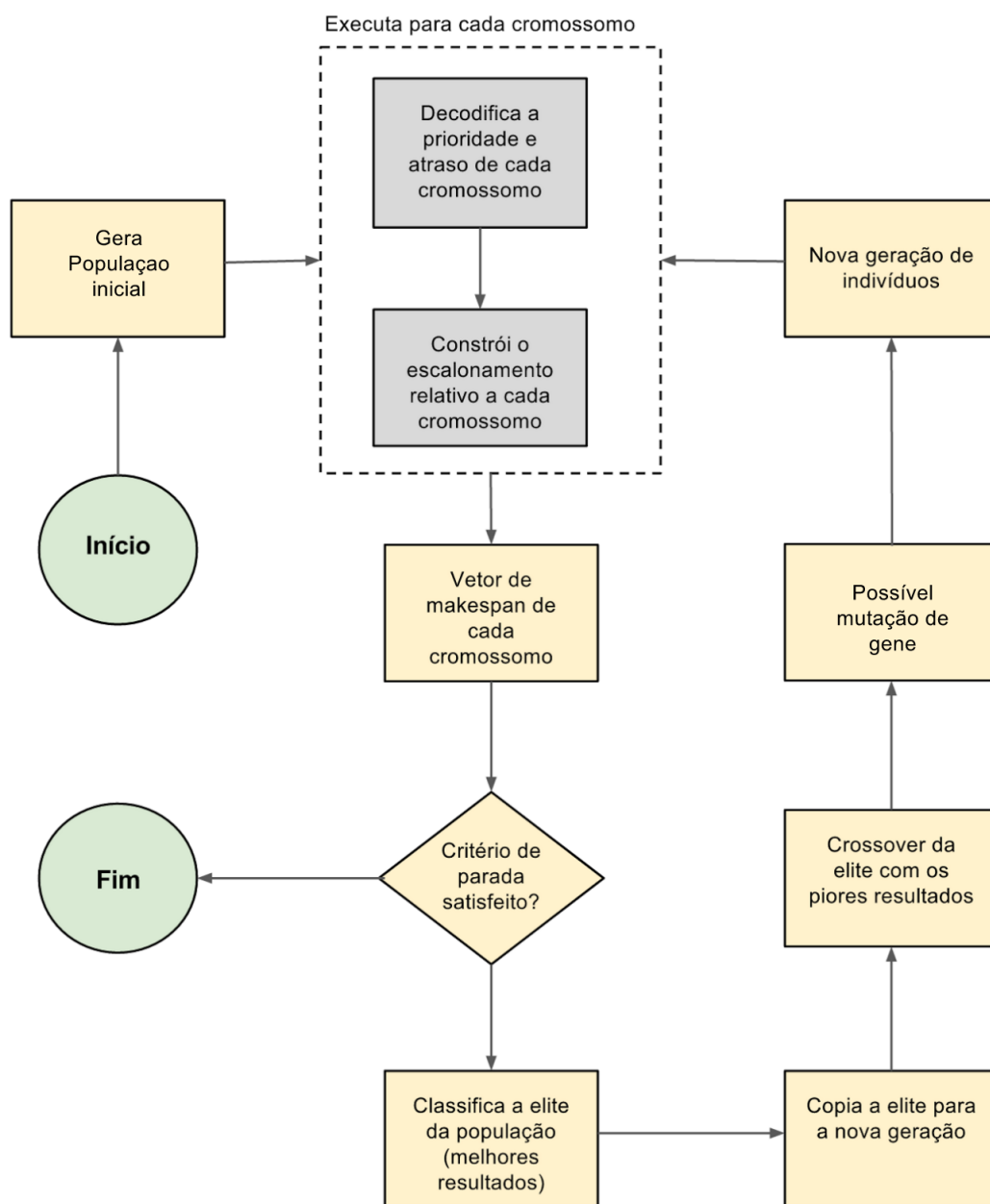


Figura 3.1 Fluxo completo do algoritmo de escalonamento de tarefas.

potencial para gerar uma boa solução e dos que estão longe da solução ideal. Essa classificação é feita ordenando o vetor de *makespans* do melhor para o pior.

7. A elite de cromossomos (os cromossomos que possuem melhor *makespan*) são copiados para gerar a próxima geração.
8. Na etapa do crossover, os cromossomos classificados como elite são utilizados para reproduzir a fim de gerar um novo indivíduo que será introduzido na população da nova geração. Essa reprodução é feita misturando aleatoriamente os genes de cada cromossomo pai.
9. Na etapa de mutação, novos cromossomos podem ser gerados aleatoriamente para serem inseridos na nova geração.
10. Finalmente, a nova geração está pronta para ser decodificada, voltando para a etapa 2 do fluxo.

O fluxo da Figura 3.1 é executado até ser atingido o critério de parada. Quando esse critério é alcançado, o algoritmo finaliza, resultando numa população com os melhores cromossomos decodificados com o tempo de finalização de cada tarefa. A melhor decodificação é escolhida para gerar o escalonamento que será usado para construir o planejamento de tarefas. Pode-se observar que as decodificações dos cromossomos de uma geração podem ser feitas em paralelo, pois um cromossomo não depende do outro para obter seu escalonamento.

3.2 Plugin do Navisworks

O software Navisworks foi usado neste trabalho como ferramenta para construção e gerenciamento de tarefas que foram usadas para construir o grafo de tarefas do algoritmo de escalonamento.

Em um primeiro momento, o modelo gráfico do projeto de construção foi feito utilizando a ferramenta CAD Revit da Autodesk. Com o Revit é possível arquitetar qualquer tipo de construção com todos os detalhes de arquitetura, estrutura, custo e tarefas. Enfim, é possível projetar todos os elementos necessários para um projeto de construção civil. Assim que o modelo gráfico ficou pronto, foi gerado um arquivo num formato que é compatível com o software Navisworks.

No Navisworks as tarefas do projeto podem ser gerenciadas a nível de escalonamento manual, que gera uma simulação de todas as etapas da construção. Essa simulação ilustra graficamente o projeto sendo executado, sempre exibindo as etapas sendo processadas e o andamento da construção.

A Autodesk, empresa que comercializa o software Navisworks, disponibiliza uma API (interface para programas de aplicações) para desenvolvedores, onde é possível criar plugins que funcionam como complemento para o software original.

O Navisworks não possui algoritmo de escalonamento de tarefas próprio, portanto, o usuário tem que escalonar todas as tarefas à mão. Tendo em vista esse problema, foi desenvolvido

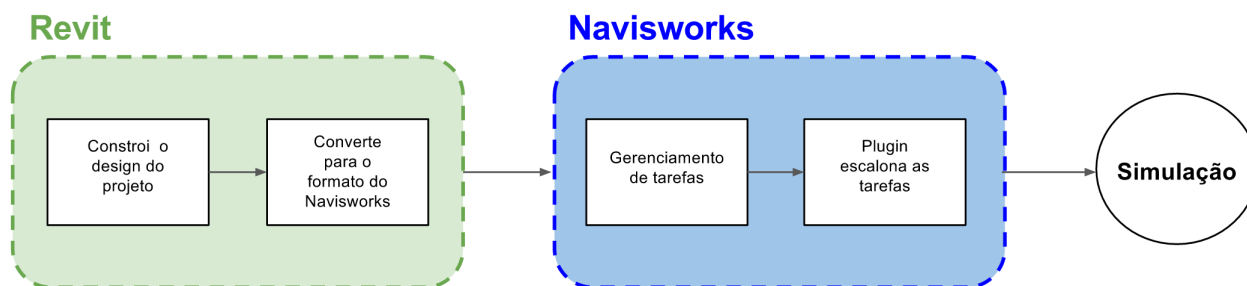


Figura 3.2 Fluxo completo do algoritmo de escalonamento de tarefas.

um plugin para a ferramenta que lê as tarefas e transforma essas tarefas em um grafo levando em conta as durações e precedência de cada uma delas. Esse grafo de tarefas é então usado como entrada para construir o algoritmo de escalonamento. Assim, os cromossomos do algoritmo genético podem ser decodificados para gerar seus *makesapns*.

Dado que software Revit permite arquitetar todo o tipo de construção e é também o mais utilizado para esse contexto de planejamento de construção civil, ele foi utilizado para construir o caso de estudo usado nos experimentos deste trabalho. O Autodesk Navisworks é o mais utilizado para complementar o Autodesk Revit na questão de planejamento e simulação da construção e por isso foi utilizado como ferramenta principal para os experimentos.

A Figura 3.2 ilustra o fluxo completo da integração entre o Revit e o Navisworks. A ferramenta CAD Revit foi usada para construção do modelo gráfico em 3D do caso de estudo, onde foi possível definir todos os elementos necessários para o projeto (estrutura, material, tarefas e custo). A partir disso, pode-se converter o projeto obtido para o formato de entrada do software BIM Navisworks e, então, dar continuidade ao planejamento do caso de estudo.

No Navisworks, as tarefas foram planejadas levando em conta o tempo de execução do projeto. Esse planejamento é feito através da ajuda de uma ferramenta do Navisworks, com a ajuda de um diagrama de Gantt, que ilustra para o usuário como ficará a execução das tarefas ao longo do tempo. Diagramas de Gantt são gráficos (ou diagramas de espaço/tempo) usados para representar diferentes etapas de um projeto [Wea06]. Porém, essa organização de tarefas é feita completamente de forma manual. Ou seja, o projetista da obra tem que planejar tudo manualmente, caso o Navisworks seja a sua ferramenta de escolha.

Portanto, o plugin entra nessa etapa onde o usuário já tem todas as tarefas definidas no Navisworks e gostaria de otimizar seu escalonamento, ou até mesmo escalonar sem nenhum planejamento prévio das datas das tarefas. O Plugin coleta apenas as informações básicas de cada tarefa (como a precedência, duração e um identificador) e encaminha para o algoritmo de escalonamento proposto. O algoritmo então calcula a solução ótima e retorna a informação para o plugin. Com as tarefas escalonadas, o plugin irá então refazer o planejamento inicial do projeto no Navisworks, atualizando as datas de início e fim para cada tarefa disponível.

Depois que as tarefas foram planejadas, uma simulação da execução do projeto em "tempo real" (a velocidade da simulação é controlada pelo usuário) foi gerada pelo software Navisworks.

3.3 Caso de Estudo

Dado que software Revit permite arquitetar todo o tipo de construção e é também o mais utilizado para esse contexto de design do modelo do projeto de construção civil, ele foi utilizado para construir o caso de estudo usado nos experimentos deste trabalho. A figura 3.3 ilustra o resultado final da construção que serviu como caso de estudo para a geração de tarefas. Esse é um design básico de uma guarita com uma arquitetura simples.

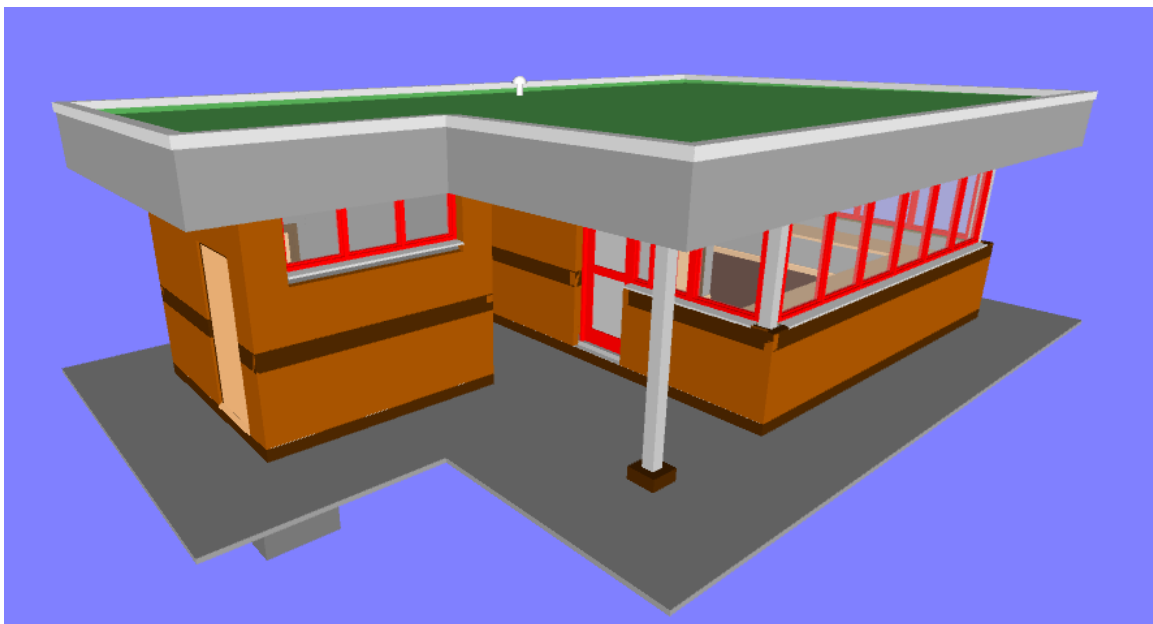


Figura 3.3 Construção feita no Revit.

Esse modelo do projeto gerado pelo Revit foi convertido para o formato do Navisworks, e então foi possível gerenciar todas as tarefas do projeto. Parte das tarefas foram geradas automaticamente, pois o Revit disponibiliza a criação de tarefas automáticas, e outra parte foi arquitetada manualmente. As tarefas finais produzidas no Navisworks podem ser vistas na Tabela 3.1 disponível nos anexos deste trabalho.

Para se obter o resultado das tarefas escalonadas, foi utilizado o plugin desenvolvido para o Navisworks. Tendo as tarefas planejadas para o projeto abertas, foi possível executar o plugin. No plugin, essas tarefas foram escalonadas pelo algoritmo proposto, até encontrar a solução ótima. Então, o plugin atualizou o planejamento das tarefas contidas no Navisworks para a solução ótima do algoritmo.

Finalmente, com o planejamento de tarefas atualizado, foi possível o vídeo da simulação onde foi mostrado o projeto sendo executado passo-a-passo.

Tabela 3.1 Tarefas planejadas à mão no Navisworks (35 tarefas).

ID	Tarefa	Duração	Recursos	Próximas Atividades	Início (Data)	Fim (Data)
0	Begin	0.0	0/0/0/0	1/2/3/4/8	01-Jun	01-Jun
1	Base1	2.0	2/3/0/0	5	01-Jun	03-Jun
2	Base2	1.0	2/3/0/0	5	01-Jun	02-Jun
3	Base3	1.0	2/3/0/0	6	07-Jun	08-Jun
4	Base4	1.0	2/3/0/0	6	07-Jun	08-Jun
5	BaseLevel2_1	2.0	2/3/0/0	7	04-Jun	06-Jun
6	BaseLevel2_2	2.0	2/3/0/0	7	09-Jun	11-Jun
7	Floor	2.0	0/0/0/5	9/10/11/12/13	12-Jun	13-Jun
8	BuyStuff	1.0	2/0/0/0	15/18/20	15-Jun	15-Jun
9	Column	1.0	0/0/0/2	30	12-Jun	12-Jun
10	Wall Front	2.0	2/3/0/0	14/15/31/25	12-Jun	13-Jun
11	Wall Right	3.0	2/4/0/0	16/32/26	14-Jun	16-Jun
12	Wall Left	3.0	2/4/0/0	17/27/34/18	17-Jun	19-Jun
13	Wall Back	2.0	2/3/0/0	33/28/19/20	20-Jun	21-Jun
14	Window Front	1.0	0/0/0/2	21	16-Jun	16-Jun
15	Door Front	1.0	0/0/0/3	30	16-Jun	16-Jun
16	Window Righth	1.0	0/0/0/3	22	17-Jun	17-Jun
17	Window Left	1.0	0/0/0/3	23	18-Jun	18-Jun
18	Door Left	1.0	0/0/0/2	30	17-Jun	17-Jun
19	Window Back	1.0	0/0/0/3	24	19-Jun	19-Jun
20	Door Back	1.0	0/0/0/5	30	20-Jun	20-Jun
21	Glass Front	1.0	0/0/2/0	30	21-Jun	21-Jun
22	Glass Right	1.0	0/0/2/0	30	23-Jun	23-Jun
23	Glass Left	1.0	0/0/2/0	30	25-Jun	5-Jun
24	Glass Back	1.0	0/0/2/0	30	27-Jun	27-Jun
25	Supp Front	2.0	2/3/0/0	29	22-Jun	22-Jun
26	Sup Righth	2.0	2/3/0/0	29	23-Jun	24-Jun
27	Sup Left	2.0	2/3/0/0	29	26-Jun	26-Jun
28	Sup Back	2.0	2/3/0/0	29	28-Jun	28-Jun
29	Roof	4.0	4/6/0/0	30	29-Jun	02-Jul
30	End	0.0	0/0/0/0		04-Jul	04-Jul
31	Details Wall Front	1.0	0/0/0/2	30	03-Jul	03-Jul
32	Details Wall Right	1.0	0/0/0/2	30	03-Jul	03-Jul
33	Details Wall Left	1.0	0/0/0/2	30	04-Jul	04-Jul
34	Details Wall Back	1.0	0/0/0/2	30	04-Jul	04-Jul

CAPÍTULO 4

Resultados e Análise

Este capítulo apresenta os resultados dos experimentos computacionais feitos com o algoritmo proposto, uma união da heurística de algoritmo genético com o construtor de escalonamento proposto por Mendes et al. [JMM05]. Os experimentos foram feitos em uma máquina com processador Intel(R) Core(TM) i5 CPU M460 @ 2.53GHz, memória RAM de 4.0GB e sistema operacional Windows 7 Ultimate 64-bits.

4.1 Configuração do Algoritmo Genético

O algoritmo genético foi utilizado para calcular o resultado ótimo, ou seja, o cromossomo que obteve a melhor decodificação que leva ao melhor escalonamento de tarefas. A tabela 4.1 mostra os valores usados para se obterem os resultados. Esses valores foram obtidos através de vários experimentos resultando na melhor configuração por escalonamento obtendo os melhores escalonamentos. Sendo assim, foram utilizados para todos os resultados dos experimentos contidos nesse capítulo.

Tabela 4.1 Parâmetros do algoritmo genético.

Tamanho da população	10 x número de atividades
Probabilidade de mutação	0.10
Melhores indivíduos copiados	20%
Critério de Parada	10

4.2 Resultados dos Escalonamentos

Foram executadas dois tipos de experimentos. Primeiro, experimentos comparando o tempo de escalonamento dos projetos feitos manualmente e também pelo algoritmo. Para o primeiro experimento foram usados projetos com 35 e 100 atividades. No segundo experimento, foram escalonadas tarefas com 250, 500 e 1000 atividades utilizando apenas o algoritmo. Para os dois experimentos, todos os projetos tiveram suas precedências bem definidas, seguindo o padrão da Tabela 3.1.

Para os testes, as durações das atividades foram definidas aleatoriamente, assim como os recursos necessário para cada uma delas. Todos os projetos utilizaram a mesma quantidade de recursos disponíveis cada um, sendo assim todos só poderiam utilizar 4 recursos diferentes.

Os projetos foram testados de forma independente para não haver concorrência de recursos. A Tabela 4.2 mostra a capacidade de cada recurso para este experimento. Porém, cada tarefa do projeto poderia utilizar a quantidade de recursos (dentro do limite da capacidade) que supria sua necessidade.

Tabela 4.2 Capacidade de recursos para o primeiro experimento.

Recursos	Capacidade Máxima
Recurso1	4 unidades
Recurso2	6 unidades
Recurso3	2 unidades
Recurso4	5 unidades

4.2.1 Primeiro experimento

Primeiramente, para o projeto descrito da Tabela 3.1 com 35 atividades e 4 recursos necessários, o algoritmo executou com os parâmetros genéticos descritos previamente e foram obtidos o escalonamento da Tabela 4.3. Pela primeira tabela, escalonada a mão antes de aplicar o algoritmo, o tempo de execução de todas as tarefas seria de 34 dias compreendidos entre os dias 01-Jun e 4-Jul. Na segunda tabela, resultado do escalonamento do algoritmo, podemos ver que a solução ótima resulta na execução do projeto em 24 dias compreendidos entre os dias 01-Jun e 24-Jun.

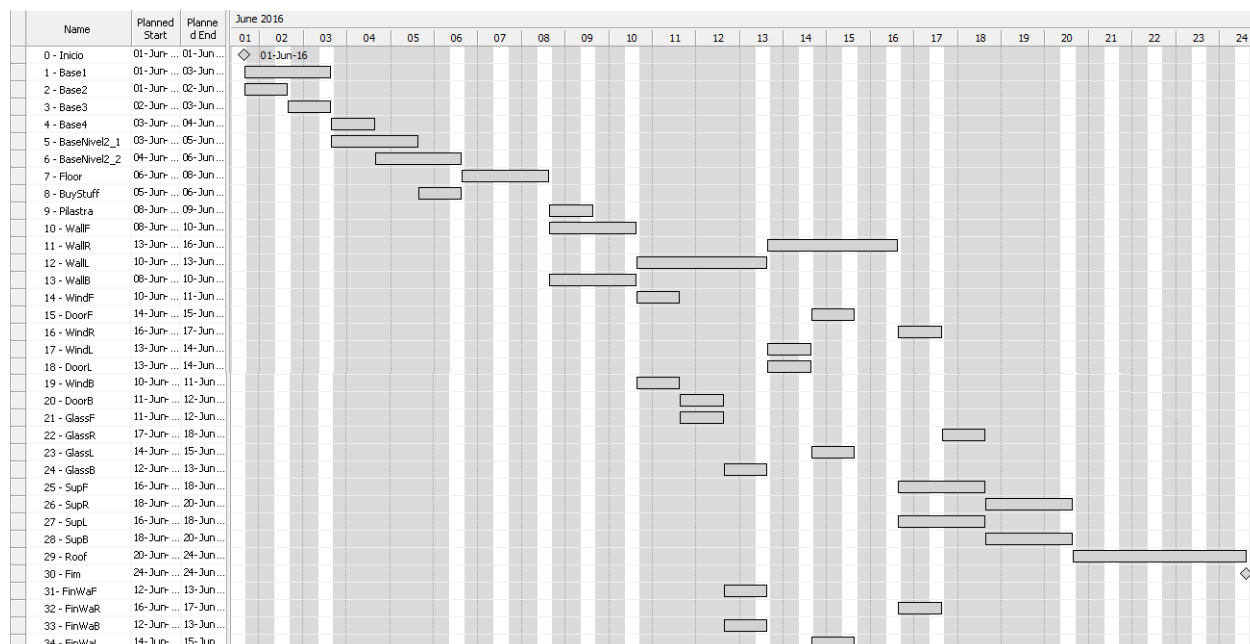


Figura 4.1 Diagrama de Gantt das tarefas escalonadas no Navisworks.

Tabela 4.3 Tarefas do Navisworks após o escalonamento (35 tarefas).

ID	Tarefa	Duração	Recursos	Próximas Atividades	Início (Data)	Fim (Data)
0	Begin	0.0	0/0/0/0	1/2/3/4/8	01-Jun	01-Jun
1	Base1	2.0	2/3/0/0	5	01-Jun	03-Jun
2	Base2	1.0	2/3/0/0	5	01-Jun	02-Jun
3	Base3	1.0	2/3/0/0	6	02-Jun	03-Jun
4	Base4	1.0	2/3/0/0	6	03-Jun	04-Jun
5	BaseLevel2_1	2.0	2/3/0/0	7	03-Jun	05-Jun
6	BaseLevel2_2	2.0	2/3/0/0	7	04-Jun	06-Jun
7	Floor	2.0	0/0/0/5	9/10/11/12/13	06-Jun	08-Jun
8	BuyStuff	1.0	2/0/0/0	15/18/20	05-Jun	06-Jun
9	Column	1.0	0/0/0/2	30	08-Jun	09-Jun
10	Wall Front	2.0	2/3/0/0	14/15/31/25	08-Jun	10-Jun
11	Wall Right	3.0	2/4/0/0	16/32/26	13-Jun	16-Jun
12	Wall Left	3.0	2/4/0/0	17/27/34/18	10-Jun	13-Jun
13	Wall Back	2.0	2/3/0/0	33/28/19/20	08-Jun	10-Jun
14	Window Front	1.0	0/0/0/2	21	10-Jun	11-Jun
15	Door Front	1.0	0/0/0/3	30	14-Jun	15-Jun
16	Window Righth	1.0	0/0/0/3	22	16-Jun	17-Jun
17	Window Left	1.0	0/0/0/3	23	13-Jun	14-Jun
18	Door Left	1.0	0/0/0/2	30	13-Jun	14-Jun
19	Window Back	1.0	0/0/0/3	24	10-Jun	11-Jun
20	Door Back	1.0	0/0/0/5	30	11-Jun	12-Jun
21	Glass Front	1.0	0/0/2/0	30	11-Jun	12-Jun
22	Glass Right	1.0	0/0/2/0	30	17-Jun	18-Jun
23	Glass Left	1.0	0/0/2/0	30	14-Jun	15-Jun
24	Glass Back	1.0	0/0/2/0	30	12-Jun	13-Jun
25	Supp Front	2.0	2/3/0/0	29	16-Jun	18-Jun
26	Sup Righth	2.0	2/3/0/0	29	18-Jun	20-Jun
27	Sup Left	2.0	2/3/0/0	29	16-Jun	18-Jun
28	Sup Back	2.0	2/3/0/0	29	18-Jun	20-Jun
29	Roof	4.0	4/6/0/0	30	20-Jun	24-Jun
30	End	0.0	0/0/0/0		24-Jun	24-Jun
31	Details Wall Front	1.0	0/0/0/2	30	12-Jun	13-Jun
32	Details Wall Right	1.0	0/0/0/2	30	16-Jun	17-Jun
33	Details Wall Left	1.0	0/0/0/2	30	12-Jun	13-Jun
34	Details Wall Back	1.0	0/0/0/2	30	14-Jun	15-Jun

O escalonamento gerado pelo algoritmo foi passado para o Navisworks através do plugin.

A Figura 4.1 ilustra a tela de planejamento de tarefas do Navisworks. Esta figura apresenta como ficaram as tarefas escalonadas já atualizadas do software Navisworks através do diagrama de Gantt (ou diagrama de espaço/tempo).

Na tabela de tarefas resultantes já possibilita ver as tarefas sendo planejadas em paralelo, porém, o diagrama mostra isso com mais clareza. O diagrama é uma vista de linha de tempo que torna mais fácil ver como um projeto está planejado. Visualizando as tarefas do projeto dessa forma, pode-se ver como elas se relacionam entre si ao longo do tempo. E então, tendo as tarefas atualizadas, foi possível ver a disponibilidade e duração de cada tarefa organizados no tempo. Por fim, o projetista poderia simular o projeto sendo construído passo a passo através de um vídeo gerado.

Os detalhes das tarefas do projeto com 100 atividades seria muito extenso para apresentar neste trabalho, por isso ele está representado na tabela contendo seu tempo final planejado manualmente e também o resultado do escalonamento. A Tabela 4.4 apresenta as diferenças nos tempos de execução para cada projeto (35 e 100). Pela tabela, é fácil notar que a diferença entre o tempo de execução de um projeto escalonado sem ferramenta e o tempo de execução do projeto escalonado pelo algoritmo cresce com a quantidade de tarefas.

Tabela 4.4 Diferenças dos tempos de execução dos projetos.

Qtd Tarefas	Qtd Recursos	Tempo antes do algoritmo	Tempo depois do algoritmo	Diferença
35	4	34 dias	23 dias	11 dias
100	4	187 dias	141 dias	46 dias

4.2.2 Segundo experimento

Durante a etapa de experimentos que fazem parte deste trabalho, não foi possível ter acesso a um projeto real com um número suficientemente grande de atividades para que fosse possível testar os algoritmos desenvolvidos de maneira mais robusta. Para contornar esse empecilho, foram criados grafos através de procedimentos pseudo-aleatórios possuindo 250, 500 e 1000 atividades. Após gerados, esses grafos foram utilizados numa segunda rodada de experimentos, cujo resultados podem ser vistos na Tabela 4.5. Ela mostra o resultado das funções fitness para o melhor cromossomo escolhido pelo algoritmo genético.

Tabela 4.5 Resultado da função fitness para cada projeto

Qtd Tarefas	Qtd Recursos	Escalonamento do algoritmo
250	4	412 dias
500	4	784 dias
1000	4	1665 dias

Conclusão

Neste trabalho foi mostrado o problema de escalonamento para construção civil, apresentando os problemas de se escalonar tarefas sem o apoio de software ou algoritmo. Em seguida foi apresentado o BIM (Building Information Modelling) e como ele pode ajudar no planejamento de tarefas, estrutura e custos da construção. Dada sua estratégia BIM e sua grande quantidade de usuários, foi escolhido o software Navisworks como ferramenta para extrair tarefas a fim de organizar o escalonamento.

Para o escalonamento foi utilizado uma heurística básica de um algoritmo genético, onde as populações de cromossomos podem sofrer reprodução, crossover e mutações a fim de criar novas gerações e encontrar o indivíduo ótimo. Se fez uso de um decodificador para escalonamento de tarefas proposto por Mendes et al. [JMM05] com o propósito de encontrar o fitness de cada cromossomo.

A abordagem sugerida foi testada em 5 projetos de tamanhos diferentes: 35, 100, 250, 500 e 1000 tarefas respectivamente. Para o primeiro experimento foram usados os projetos de 35 e 100 tarefas, comparando os resultados obtidos pelo escalonamento manual e pelo algoritmo. Foi demonstrado que a medida que o número de tarefas aumenta, também aumenta a complexidade do projeto, fazendo com que o tempo desperdiçado entre uma tarefa e outra seja muito maior do que deveria. Para o segundo experimento, foram utilizados os projetos de 250, 500 e 1000 atividades a fim de demonstrar o poder computacional do algoritmo.

Com o algoritmo proposto foi possível obter um planejamento otimizado, respeitando as necessidades e capacidades dos recursos, assim como evitando o tempo ocioso. O planejamento original foi então substituído pelo escalonamento otimizado no Navisworks através do plugin desenvolvido para coletar e refazer o planejamento do projeto.

Este trabalho considera que a construção contém apenas um projeto. Porém, na prática uma construção pode ser vários tipos de projetos: estrutura, eletricidade, encanamento, ambiente, entre outros. Trabalhos futuros incluem a possibilidade de escalonar diferentes projetos contidos numa construção com o objetivo de paralelizar ainda mais a execução desses trabalhos.

Referências Bibliográficas

- [4011] A report for the government construction client group, 2011.
- [AB14] D. Bordoli A. Baldwin. Handbook for construction planning and scheduling. 2014.
- [Aut12a] Inc. Autodesk. Autodesk navisworks, 2012.
- [Aut12b] Inc. Autodesk. Autodesk revit, 2012.
- [BH98] G. Ballard and G Howell. Shielding production: an essential step in production control. *Journal of Construction Engineering Management*, 1998.
- [Bla83] Lenstra J.K. Rinnooy Kan A. H.G. Blazewicz, J. Scheduling subject to resource constraints: Classification and complexity. *Discrete Applied Mathematics*, 1983.
- [Cro12] Ray Crotty. The impact of building information modelling: Transforming construction. *London: SPON/Routledge*, 2012.
- [CT87] Alvarez-Valdes R. Christofides, N. and J.M. Tamarit. Problem scheduling with resource constraints : A branch and bound approach. *European Journal of Operational Research*, 1987.
- [DDM06] R.Leus D. Debels, B. De Reyck and M.Vanhoucke. A hybrid scatter search/electromagnetism meta- heuristic for project sheduling. *European Journal of Operational Research*, 2006.
- [DV05] D. Debels and M. Vanhoucke. A decomposition-based heuristic for the resource-constrained project scheduling problem. *Faculty of Economics and Business Administration, University of Ghent*, 2005.
- [EL11] Teicholz P. Sacks R. Eastman, C. and K. Liston. *BIM Handbook (Second Edition)*. Wiley, 2011.
- [FH04] K. Fleszar and K.S. Hindi. Solving the resource-constrained project scheduling problem by a variable neighbourhood search. *European Journal of Operational Research*, 2004.
- [Gol89] D. E. Goldberg. *Genetic Algorithms in Search, Optimization Machine Learning*. Addison-Wesley, 1989.

- [JMM05] José F. Gonçalves Jorge M. Mendes, Maurício G. C. Resende. A random key based genetic algorithm for the resource constrained project scheduling problem. *ATT Labs Research Technical Report TD-6DUK2C*, 2005.
- [KH99] R. Kolisch and S. Hartmann. Heuristic algorithms for solving the resource-constrained project scheduling problem: Classification and computational analysis. *Handbook on Recent Advances in Project Scheduling*, 1999.
- [Kos02] G.; Ballard G.; Tommelein I. Koskela, L.; Howell. Foundations of lean construction. *Elsevier*, 2002.
- [KS03] Y. Kochetov and A. Stolyar. Evolutionary local search with variable neighborhood for the resource constrained project scheduling problem. *Proceedings of the 3rd International Workshop of Computer Science and Information Technologies*, 2003.
- [Loc74] K.G. Lockyer. *An Introduction to Critical Path Analysis*. Pitman, London, 1974.
- [MG07] J.J.M. Mendes and J.F. Goncalves. A memetic algorithm-based heuristics for the resource constrained project scheduling problem. *Proceedings of II International Conference on Computational Methods for Coupled Problems in Science and Engineering*, 2007.
- [MKR04] U. Breskvar M. Kljajc and B. Rodic. Computer aided scheduling with use of genetic algorithms and a visual discrete event simulation model. *WSEAS Transactions on Systems*, 2004.
- [MM08] J. Magalhães-Mendes. Project scheduling under multiple resources constraints using a genetic algorithm. *WSEAS TRANSACTIONS on BUSINESS and ECONOMICS*, 2008.
- [MO97] Askew W. Mawdesley, M. and M. O'Reilly. *Planning and Controlling Construction Projects*. Addison Wesley Longman and The Chartered Institute of Building, 1997.
- [MPM04] C. Artigues M. Palpant and P. Michelon. Lssper: Solving the resource-constrained project scheduling problem with large neighbourhood search. *Annals of Operations Research*, 2004.
- [Nar08] K. Lalit Narayan. Computer aided design and manufacturing. *New Delhi: Prentice Hall of India*, 2008.
- [Pat04] C. Patrick. *Construction Project Planning and Scheduling*. PEARSON Prentice Hall, 2004.
- [PB99] R. Mohring K. Neumann E. Pesch P. Brucker, A Drexl. Resource-constrained project scheduling: Notation, classification, models and methods. *European Journal of Operational Research*, 1999.

- [PY03] H. Pan and C.H. Yeh. Ga for fuzzy multi-mode resource-constrained project scheduling. *WSEAS Transactions on Systems*, 2003.
- [Ran08] M. Ranjbar. Solving the resource-constrained project scheduling problem using filter-and-fan approach. *Applied Mathematics and Computation*, 2008.
- [RV08] Christophe Cruz Renaud Vanlande, Christophe Nicolle. Ifc and building lifecycle management. *Elsevier*, 2008.
- [Sed06] M. Seda. Solving resource-constrained project scheduling problem as a sequence of multi-knapsack problems. *WSEAS Transactions on Information Science Applications*, 2006.
- [Smi07] Deke Smith. An introduction to building information modeling (bim). *Journal of Building Information Modeling*, 2007.
- [Smi14] Peter Smith. Bim the 5d project cost manager. *Procedia - Social and Behavioral Sciences*, 2014.
- [TR14] R. F. TOSO and M. G.C. RESENDE. A c++ application programming interface for biased random-key genetic algorithms. *ATT Labs Research Technical Report*, 2014.
- [VVQ03] F. Ballestin V. Valls and M.S. Quintanilla. A hybrid genetic algorithm for the rcpsp. *Technical report, Department of Statistics and Operations Research*, 2003.
- [VVQ05] F. Ballestin V. Valls and M.S. Quintanilla. Justification and rcpsp: A technique that pays. *European Journal of Operational Research*, 2005.
- [Wea06] Patrick Weaver. A brief history of scheduling - back to the future. *myPrimavera Conference*, 2006.
- [WJ96] J.P. Womack and D.T. Jones. *Lean Thinking: Banish Waste and Create Wealth in Your Corporation*. Simon and Schuster, 1996.
- [WR90] Jones D.T. Womack, J.P. and D. Roos. *The Machine that Changed the World: The Story of Lean Production*. HarperCollins, 1990.
- [YP05] C.H. Yeh and H. Pan. System development for fuzzy project scheduling. *WSEAS Transactions on Business and Economics*, 2005.
- [11] J.F. Gon calves and M.G.C. Resende. Biased random-key genetic algorithms for combinatorial optimization. *Journal of Heuristics*, 2011.

