
UMA METAHEURÍSTICA PARA A PROGRAMAÇÃO DE PROJECTOS COM MULTI-MODOS E RECURSOS LIMITADOS

A METAHEURISTIC FOR PROJECT SCHEDULING WITH MULTI- MODES AND LIMITED RESOURCES

Prof. Jorge Magalhães-Mendes, Ph.D. Eng.

Doutor em Engenharia Mecânica
Departamento de Engenharia Civil
Instituto Superior de Engenharia do Porto
Instituto Politécnico do Porto
Rua Dr. António Bernardino de Almeida 431, 4200-072
Porto, Portugal
jjm@isep.ipp.pt

RESUMO: Este artigo apresenta uma nova abordagem (MM-GAV-FBI), aplicável ao problema da programação de projectos com restrições de recursos e vários modos de execução por actividade, problema conhecido na literatura anglo-saxónica por MRCPS. Cada projecto tem um conjunto de actividades com precedências tecnológicas definidas e um conjunto de recursos limitados, sendo que cada actividade pode ter mais do que um modo de realização. A programação dos projectos é realizada com recurso a um esquema de geração de planos (do inglês Schedule Generation Scheme - SGS) integrado com uma metaheurística. A metaheurística é baseada no paradigma dos algoritmos genéticos. As prioridades das actividades são obtidas a partir de um algoritmo genético. A representação cromossómica utilizada baseia-se em chaves aleatórias. O SGS gera planos não-atrasados. Após a obtenção de uma solução é aplicada uma melhoria local. O objectivo da abordagem é encontrar o melhor plano (*planning*), ou seja, o plano que tenha a menor duração temporal possível, satisfazendo as precedências das actividades e as restrições de recursos. A abordagem proposta é testada num conjunto de problemas retirados da literatura da especialidade e os resultados computacionais são comparados com outras abordagens. Os resultados computacionais validam o bom desempenho da abordagem, não apenas em termos de qualidade da solução, mas também em termos de tempo útil.

Palavras-chave: Gestão de projectos. Planeamento. Programação. Metaheurísticas. *MRCPS*.

ABSTRACT: As the complexity of projects increases, the requirement of an organized planning and scheduling process is enhanced. The need for organized planning and scheduling of a construction project is influenced by a variety of factors (e.g., project size and number of project activities). To plan and schedule a construction project, activities must be defined sufficiently. The level of detail determines the number of activities contained within the project plan and schedule. So, finding feasible schedules which efficiently use scarce

resources is a challenging task within project management. In this context, the well-known Resource Constrained Project Scheduling Problem (*RCPS*) has been studied during the last decades. In the *RCPS* the activities of a project have to be scheduled such that the makespan of the project is minimized. So, the technological precedence constraints have to be observed as well as limitations of the renewable resources required to accomplish the activities. Once started, an activity may not be interrupted. This problem has been extended to a more realistic model, the multi-mode resource constrained project scheduling problem (*MRCPS*), where each activity can be performed in one out of several modes. Each mode of an activity represents an alternative way of combining different levels of resource requirements with a related duration. Each renewable resource has a limited availability such as manpower and machines for the entire project. The objective of the *MRCPS* problem is minimizing the *makespan*. While the exact methods are available for providing optimal solution for small problems, its computation time is not feasible for large-scale problems. This paper presents a genetic algorithm-based approach (*MM-GAV-FBI*) for the multi-mode resource constrained project scheduling problem. The idea of this new approach is integrating a genetic algorithm with a schedule generation scheme. This study also proposes applying a local search procedure trying to improve the initial solution. The chromosome representation of the problem is based on random keys. The schedule is constructed using a schedule generation scheme (SGS) in which the priorities of the activities are defined by the genetic algorithm. The experimental results of *MM-GAV-FBI* on project instances show that this approach is an effective method for solving the *MRCPS*.

Keywords: Project management. Planning. Scheduling. Metaheuristics. *MRCPS*.

1. INTRODUÇÃO

O problema de programação de projectos com recursos limitados (na terminologia anglo-saxónica *Resource Constrained Project Scheduling Problem - RCPS*) tem vindo a levantar o maior interesse por parte de investigadores em todo o mundo.

O *RCPS* consiste em definir o instante de início e fim de cada uma das actividades, assim como os recursos a associar à execução de cada uma dessas actividades. Os recursos são limitados e têm de ser alocados às actividades de modo a permitir a conclusão do projecto, tendo em conta um determinado objectivo de optimização, normalmente a minimização da duração do projecto.

Na indústria da construção, e em particular na realização de grandes projectos, é fundamental obter um plano de obra com o menor prazo possível respeitando as restrições tecnológicas e as restrições de recursos. O investimento em modelos de planeamento e programação são, por isso, de grande importância para tornar esta indústria competitiva.

Refira-se também que na indústria da construção a existência de estaleiros para a realização das actividades de uma obra implica a deslocação de pessoal e equipamentos com custos significativos por período, pelo que se justifica a melhor utilização dos meios

disponíveis.

A diversidade dos problemas na programação (também designada em português por *planificação, planeamento operacional* ou *planeamento de obra*) de projectos com recursos limitados pode ser definida através de uma combinação de diferentes parâmetros. Um parâmetro fundamental tem a ver com a definição das próprias actividades que definem o projecto. Uma actividade poderá ter apenas um único modo de execução (problemas do tipo modo único ou *single-mode*), ou vários modos de execução (problemas do tipo multi-modo ou *multi-mode*). Outro parâmetro possível consiste no tipo de relação que existe entre a duração de uma actividade e o nível de recursos aplicados na sua execução.

O problema de programação poderá ainda ser de natureza estocástica, baseado em informação à qual se associa uma determinada probabilidade de ocorrência, ou de natureza determinística, se toda a informação for perfeitamente conhecida.

No que se refere ao tipo de função objectivo definida para o projecto e que representa o objectivo geral que se pretende obter ao resolver um problema, a minimização da duração total do projecto (*makespan*) tem sido o mais estudado até agora. Em problemas com este tipo de função objectivo, procura-se planear as actividades do projecto de modo a garantir que o projecto termine o mais cedo possível.

No presente trabalho apresenta-se uma metaheurística para a resolução do problema da programação de projectos com actividades do tipo multi-modo (*Multi-mode Resource Constrained Project scheduling Problem - MRCPSP*), baseada num algoritmo genético. A medida de optimização a utilizar é a minimização da duração do projecto.

2. REVISÃO DA LITERATURA

No presente artigo, o interesse maior é ao nível da programação. Este nível está relacionado com as suposições assumidas sobre os modos (vários modos possíveis para uma mesma actividade), relações de precedência, relações de dependência (sobreposição ou não), representação lógica (tarefas nos nós ou nos arcos) das actividades, bem como informações relativas aos recursos (procura, disponibilidade).

A programação, para o problema *MRCPSP*, determina o modo em que cada actividade deve ser realizada, isto é, qual a melhor opção recurso-duração deve ser utilizada e quando cada actividade deve ser iniciada e terminada.

Blazewicz et al. (1983) demonstraram que o problema *RCPS*P é uma generalização do problema clássico do *JSP* (*Job Shop Problem*) o qual pertence à classe de problemas do tipo

NP-difícil. Assim sendo o problema *MRCPSP*, sendo uma generalização do *RCPSP*, é também um problema NP-difícil.

A grande maioria de procedimentos de programação, podem ser divididos em dois principais grupos, baseados nas abordagens empregues: o primeiro, e de longe a maior categoria, denominado de procedimentos heurísticos procuram produzir soluções de boa qualidade. O segundo grupo consiste nos procedimentos que procuram produzir a melhor solução ou solução óptima. Estes são classificados como procedimentos óptimos, também designados exactos ou analíticos porque geralmente envolvem alguma forma de programação matemática ou outro procedimento analítico mais rigoroso.

Dentro de cada um destes grupos há possíveis esquemas de divisão. Os procedimentos heurísticos, por exemplo, podem ser divididos em duas classes (STORER ET AL., 1992).

1. Heurísticas específicas, desenvolvidas para cada tipo de problema, baseadas nas especificações da questão tratada, como por exemplo, regras de prioridade e abordagens de sistemas especialistas;
2. Metaheurísticas, que são formas alternativas de busca do espaço de soluções baseadas no princípio da vizinhança, como por exemplo, *simulated annealing* (SA), algoritmos genéticos (AG) e *tabu search*.

Em relação aos procedimentos exactos, outras duas classes principais podem ser identificadas (MACCARTHYY; LIU, 1993).

1. *Métodos exactos eficientes*: são métodos que geram soluções óptimas em tempo polinomial;
2. *Métodos exactos enumerativos*: são métodos que envolvem uma enumeração parcial do conjunto de todas as soluções possíveis. Esta abordagem é desenvolvida baseada em dois conceitos principais (DAY; HOTTENSTEIN, 1970): a) o uso de uma técnica de enumeração controlada para considerar todas as soluções possíveis e, b) a eliminação de potenciais soluções inaceitáveis. As mais gerais são formulações de programação matemática, métodos de *branch and bound* e métodos de eliminação.

Devido ao crescimento exponencial do tempo computacional em função do tamanho do problema, a geração de soluções óptimas globais, ainda para problemas reais de pequena dimensão, permanece computacionalmente impraticável através de métodos de enumeração completa (DAVIS, 1973), pelo que procedimentos baseados em heurísticas têm sido

largamente utilizados como abordagens de solução ao problema. Estes tipos de abordagem são hoje, as formas mais eficientes de obter uma solução com qualidade para o problema, em tempo computacional aceitável (BOCTOR, 1990).

Por comparação com os trabalhos de investigação no problema RCPSP (ver por exemplo (RANJBAR, 2008; MENDES ET AL., 2009; KOLISCH; HARTMANN, 2006), são menos os trabalhos que abordam o caso de múltiplos modos MRCPS. Alguns dos trabalhos referidos pela literatura são os propostos por Slowinski (1981); Talbot (1982); Patterson et al. (1990); Boctor (1990); Li e Wills (1992); Drexel e Gruenewald (1993); Moselhi e Lorterapong (1993); Boctor (1993); Vaca (1995); Boctor (1996); Mori e Tseng (1997); Kolisch e Drexl (1997); De Reyck e Herroelen (1999); Heilmann (2001); Hartmann (2001); Damak et al. (2009); Lova e Tormos (2009).

Slowinski (1981) apresenta duas abordagens baseadas na programação linear multi-objetivos, que tratam do caso de múltiplos modos com utilização de diferentes categorias de recursos (renováveis, não renováveis e com dupla restrição) e múltiplos critérios de optimização, levando em consideração a interrupção de actividades.

Talbot (1982), introduz métodos para formular e resolver o problema de programação de projectos onde a duração é função dos recursos. O autor desenvolveu uma metodologia de dois estágios que utiliza regras heurísticas no primeiro estágio e um algoritmo de enumeração implícita no segundo estágio, procurando minimizar a duração do projecto. Este mesmo algoritmo é utilizado para abordar o problema de compressão de projectos com limitação de recursos.

Patterson et al. (1990), apresenta uma experiência computacional para resolver a versão de múltiplos modos do problema de programação com restrição de recursos procurando minimizar a duração do projecto e maximizar o valor presente. Um procedimento de procura em profundidade baseada no algoritmo de *branch and bound* é utilizado como metodologia de solução. Este modelo consiste de duas fases: inicialização do problema de enumeração e geração do espaço de solução. Na fase de inicialização uma heurística dá as prioridades das actividades e a selecção do modo de execução das actividades. Este procedimento guia a procura da solução inicial, afectando a ordem na qual as actividades e os modos serão considerados na atribuição dos recursos durante a fase de enumeração. Esta segunda fase é utilizada para gerar o espaço de solução e para avaliar todas as soluções parciais que podem ser estendidas para uma solução óptima.

Boctor (1990), desenvolveu um esquema geral para resolver problemas de tamanho

grande, para os quais a duração das actividades é função dos recursos alocados à sua execução, procurando minimizar a duração total do projecto. Foi realizado um estudo comparativo do desempenho de 21 regras heurísticas. O modelo aplica os conceitos tradicionais de prioridades de actividades e selecção de modos.

Jones (1992), usou uma abordagem de Monte Carlo para aleatoriamente gerar um grande conjunto de programações viáveis para de seguida seleccionar a de melhor desempenho.

Li e Willis (1992), apresentam um procedimento heurístico onde um projecto é programado do início para o fim e do fim para o início iterativamente até que não sejam obtidas melhorias na duração do projecto. Esta programação iterativa aborda diferentes categorias e tipos de recursos, com diferentes níveis de utilização, tais como: recursos renováveis constantes no tempo, recursos renováveis variáveis no tempo, recursos não renováveis e duplamente restritos constantes no tempo, recursos não renováveis e duplamente restritos.

Drexel e Gruenewald (1993), apresentam um método de programação estocástico para formular e resolver o caso de programação com múltiplos modos e restrição de recursos, bem como a compressão de projectos com múltiplos modos. A contribuição do modelo é generalizar regras de programação determinísticas apropriadas incorporando uma técnica de selecção aleatória ponderada.

Moselhi e Lorterapong (1993), desenvolveram uma heurística que tem como objectivo incrementar a consistência do desempenho da tradicional regra da folga total mínima. O modelo é baseado na alocação de recursos por grupo. Os conjuntos de actividades são geradas considerando inicialmente todas as possíveis combinações das actividades activas em conflito e procurando a redução para um espaço das soluções viáveis, com prioridade pelo menor impacto sobre a duração do projecto.

Boctor (1993), publicou um estudo onde analisou sete regras de prioridade para a selecção das actividades combinando-as com três regras de prioridade para a selecção dos modos. O total de combinações possíveis ascende a 21, podendo cada uma delas dar origem a uma heurística diferente. Estas regras foram aplicadas sobre 240 problemas divididos em grupos de 50 e 100 actividades por projecto. Cada projecto pode utilizar até 4 recursos.

Vaca (1995), no seu trabalho de doutoramento apresenta um modelo que utiliza os algoritmos genéticos como abordagem heurística para resolver o problema de alocação de recursos limitados em ambiente de múltiplos modos. Além disto, o conceito de nivelamento

de recursos é incorporado no procedimento como um mecanismo guia na busca de uma solução de mínima duração, com padrões de flutuação dos recursos utilizados de alguma forma reduzidos.

Boctor (1996), propõe um algoritmo em que a programação das actividades em cada instante de tempo seja realizada com base em blocos ou conjuntos de actividades, programando o bloco que tiver o valor mais favorável.

Mori e Tseng (1997), propõem um algoritmo genético com incorporação de informação específica do problema através de uma representação específica dos operadores genéticos.

Kolish e Drexl (1997), desenvolveram uma heurística baseada em procura local e que considera os níveis de recursos não renováveis.

De Reyck e Herroelen (1999), propuseram um método de procura local e outro de pesquisa tabu para resolver o *MRCPSP* com diversas restrições adicionais. A sua abordagem divide o problema em duas partes. Depois de ter determinado uma afectação possível dos modos às actividades, prossegue com a programação das actividades resolvendo um problema clássico do tipo *RCPS*. Entre os métodos que testaram, os melhores resultados foram obtidos com a pesquisa *tabu*. Um dos inconvenientes dos métodos propostos com base nas experiências computacionais foi o facto de algumas vezes os algoritmos falharem em encontrar soluções válidas para as instâncias.

Heilmann (2001), apresentou um algoritmo para o caso mais geral do *MRCPSP* (para além de se verificar a existência de tempos mínimos entre os inícios das actividades, existem tempos máximos entre o início das mesmas) baseado em 3 regras de prioridade para a selecção das actividades e 3 regras de prioridade para a selecção dos respectivos modos.

Hartmann (2001), apresentou um algoritmo genético baseado numa lista de actividades. Após determinar uma solução viável aplica uma melhoria local.

Damak et al. (2009), propõem um algoritmo evolucionário aplicando o conceito de evolução diferencial.

Lova et al. (2009), apresentam um algoritmo genético híbrido. O algoritmo genético permite a obtenção de uma solução inicial. Sobre esta solução inicial aplicam uma estratégia de melhoria local baseada numa estratégia *forward-backward improvement* (FBI).

De seguida apresenta-se a formulação do problema.

3. FORMULAÇÃO DO PROBLEMA

O problema *MRCPSP* pode ser descrito da seguinte forma: um projecto consiste num

conjunto de actividades $J = \{0, 1, 2, \dots, n, n+1\}$ a programar. As tarefas 0 e $n+1$ são fictícias, têm duração nula e correspondem ao início e fim do projecto. As tarefas estão inter-relacionadas por dois tipos de restrições:

- Restrições de precedência que implicam que cada actividade j só pode ser programada após todas as suas actividades precedentes, P_j , terem sido concluídas;
- Restrições de capacidade que implica que uma actividade só pode ser programada se houver recursos disponíveis.

No problema *MRCPS* considerado existem K tipos de recursos renováveis, representados pelo conjunto $K = \{1, \dots, k\}$. Cada actividade tem uma duração determinística definida por p_j . Entre os instantes de início e de fim, cada actividade j necessita de $r_{j,k}$ de cada recurso k e não pode ser interrompida. Cada tipo de recurso renovável dispõe de uma capacidade limitada R_k no período de planeamento. Os parâmetros p_j , $r_{j,k}$ e R_k são determinísticos; para as actividades de início e fim do projecto $p_0 = p_{n+1} = 0$ e $r_{0,k} = r_{n+1,k} = 0$ para qualquer recurso k .

Cada actividade pode ser executada num modo a escolher entre vários modos. Um modo representa uma combinação diferente de recursos com uma duração associada. Uma vez uma actividade iniciada num dos modos de execução não pode haver nenhuma alteração de modo. Uma actividade j pode ser executada em m_j modos possíveis dados pelo conjunto $M_j = \{1, \dots, m_j\}$. O tempo de processamento ou duração da actividade j em modo m_j é dado por p_{jm} . Adicionalmente uma actividade j executada em modo m_j utiliza r_{jmk} unidades do recurso renovável k em cada período ao longo da sua execução, sendo que $r_{jmk} \leq R_k$ para cada recurso renovável k .

A resolução do problema do *MRCPS* consiste em determinar os tempos de início e de fim e seleccionar o modo de execução de cada actividade, respeitando as relações de precedência e de capacidade disponível de cada recurso, por forma a que a duração total do projecto, *makespan*, seja minimizada.

Uma formulação matemática para este tipo de problema é proposta por Talbot (1982).

4. ALGORITMOS GENÉTICOS

Os algoritmos genéticos são baseados nos mecanismos de selecção natural e da genética e foram introduzidos por Holland (1975). Os algoritmos genéticos são normalmente usados para resolver problemas de pesquisa e optimização.

Na teoria da evolução natural sobrevivem os indivíduos mais aptos. A cada indivíduo (cromossoma) de uma população está associado um conjunto de genes, que representa o seu valor ou mérito. Quanto mais apto estiver um indivíduo para a sobrevivência maior será o seu mérito.

Os mecanismos naturais de preservação e evolução de uma população são a sua reprodução e mutação. A reprodução tem por base o cruzamento genético. Através de mutação dos indivíduos é possível incluir diversidade numa espécie.

Os algoritmos genéticos utilizam os seguintes mecanismos:

- **Seleção:** consiste em seleccionar alguns dos melhores cromossomas para a população seguinte;
- **Cruzamento:** consiste no cruzamento de cromossomas preferencialmente de elevado mérito;
- **Mutação:** consiste em incluir diversidade numa população através da alteração dos genes de alguns cromossomas.

Na Figura 1 apresenta-se o pseudo-código utilizado no algoritmo genético.

Algoritmo Genético

```
{ Gerar população inicial  $S_t$ 
  Avaliar população  $S_t$ 
  Enquanto critério de paragem não satisfeito Repetir
  { Seleccionar elementos de  $S_t$  a colocar em  $S_{t+1}$ 
    Cruzar elementos de  $S_t$  e colocar em  $S_{t+1}$ 
    Mutação de novos elementos e colocar em  $S_{t+1}$ 
    Avaliar nova população  $S_{t+1}$ 
     $S_t = S_{t+1}$  }
```

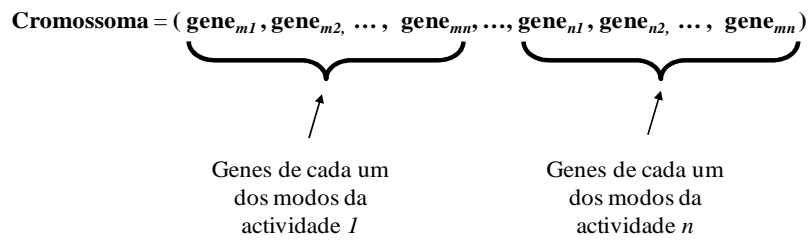
Figura 1 – Pseudo-código de um algoritmo genético standard

Fonte: Mendes (2003)

4.1 Representação cromossómica

O algoritmo genético utilizado usa chaves aleatórias como método de representação (codificação) de soluções em vez da codificação binária tradicional. Esta metodologia consiste na atribuição de números aleatórios, compreendidos entre 0 e 1, aos genes que constituem os cromossomas.

Na abordagem apresentada neste artigo cada *cromossoma inicial* é composto por mn genes:



Para cada actividade j correspondem um conjunto de modos de execução:

$$\text{Actividade } 1 = (\text{gene}_{m1}, \text{gene}_{m2}, \dots, \text{gene}_{mn})$$

$$\text{Actividade } 2 = (\text{gene}_{m1}, \text{gene}_{m2}, \dots, \text{gene}_{mn})$$

.....

$$\text{Actividade } n = (\text{gene}_{m1}, \text{gene}_{m2}, \dots, \text{gene}_{mn})$$

Mas apenas cada um destes modos, m_j , participa na construção da solução. Após a escolha do modo m_j de cada uma das actividades j o cromossoma (solução) é composto por n genes, ver Figura 2.

Actividade 1		Actividade j		Actividade n
Modo m_2	Modo m_5	Modo m_2
Gene m_2		Gene m_5		Gene m_2

Figura 2 – Exemplo de um cromossoma (solução) com n genes

Fonte: Autor

4.2 Descodificação das prioridades

Os n genes são usados para obter as prioridades de cada actividade de acordo a seguinte expressão (1):

$$\text{Prioridade}_j = \frac{DMC_j}{CCP} \times \left[\frac{1 + \text{gene}_{mj}}{2} \right] \quad j = 1, \dots, n \quad (1)$$

onde,

DMC_j é a duração do mais longo caminho desde o início da actividade j até ao fim do projecto;

CCP é o valor da duração do caminho crítico do projecto;

m_j é o gene do modo seleccionado que corresponde à actividade j .

4.3 Estratégia evolutiva

Inicialmente é gerada aleatoriamente uma população de cromossomas. Esta população será posteriormente transformada através da aplicação dos operadores genéticos. A

reprodução é conseguida usando uma estratégia elitista (GOLDBERG, 1989) na qual os melhores indivíduos de uma população são copiados para a geração seguinte. A vantagem desta estratégia é garantir que a melhor solução vá melhorando monotonicamente de geração para geração.

Neste trabalho o método de selecção da roleta é utilizado para a reprodução. A característica da selecção da roleta é estocástica. A probabilidade de sobrevivência de cada indivíduo é determinada pelo seu mérito. No método da selecção através da roleta são representadas as probabilidades de sobrevivência de todos os elementos da população. De seguida a roleta é accionada por várias vezes (GOLDBERG, 1989).

Após a selecção dos indivíduos para reprodução, o operador de cruzamento é aplicado da seguinte forma:

1. Selecciona-se para cada cromossoma inicial um par para reprodução, aplicando novamente a selecção através da roleta;
2. Selecciona-se aleatoriamente uma posição k no cromossoma inicial entre a posição 1 e o comprimento do cromossoma l . Dois novos cromossomas são obtidos trocando todos os genes entre as posições $k+1$ e l [1], ver Figura 3.

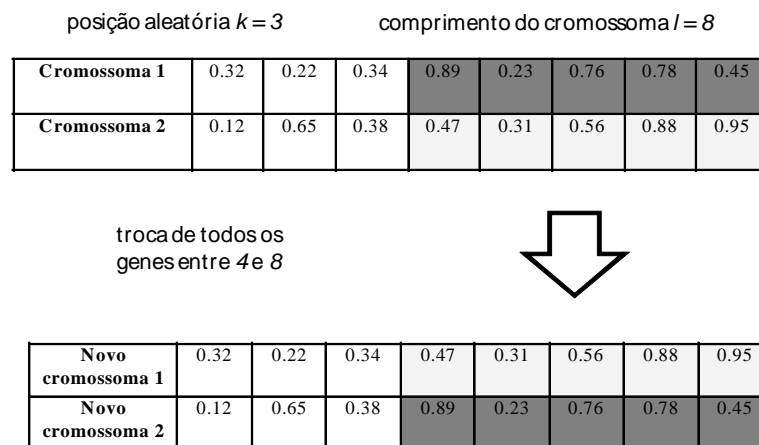


Figura 3 – Exemplo de aplicação do operador de cruzamento
Fonte: Magalhães-Mendes (2008)

O operador de mutação é aplicado numa população com uma probabilidade predeterminada. Neste trabalho a probabilidade de mutação é de 0.001. A título de exemplo, numa população de 400 cromossomas, espera-se que possam ser alterados $400 \cdot 240 \cdot 0.001 = 96$ genes, por geração. Este processo evita a convergência prematura de uma população.

No caso particular dos cromossomas cuja reprodução foi obtida por selecção elitista não é aplicada mutação.

5. TIPOS DE PLANOS

Existem os seguintes tipos de planos:

- *Plano semi-activo*: é um plano admissível que assegura que cada actividade é iniciada o mais cedo possível (isto é, não existem tempos de inactividade desnecessários);
- *Plano activo*: é um plano admissível que assegura que nenhuma actividade pode ser iniciada mais cedo sem provocar um atraso noutras tarefas ou violar as restrições tecnológicas ou de recursos. Os planos activos formam um subconjunto dos semi-activos, isto é, um plano activo é necessariamente semi-activo. De notar que a solução óptima pertence ao subconjunto dos activos;
- *Plano não-Atrasado*: é um plano admissível que assegura que nenhum recurso fica inactivo quando pode iniciar o processamento de alguma actividade. Estes planos fazem parte do conjunto dos activos, logo, também são semi-activos.

Neste trabalho são gerados planos não-atrasados obtidos através de um SGS.

6. ESQUEMA DE GERAÇÃO DE PLANOS

O SGS utilizado para obter planos não-atrasados é descrito em detalhe no trabalho (MENDES, 2003). As prioridades das actividades utilizadas pelo SGS são as obtidas pelo algoritmo genético.

7. PESQUISA LOCAL

Os algoritmos de pesquisa local movem-se de solução em solução no espaço de pesquisa até ser obtido um valor óptimo ou um critério de paragem ser alcançado. Neste trabalho é aplicado um algoritmo de pesquisa local através da adaptação do mecanismo proposto por Klein (2000), com vista ao melhoramento da solução obtida através do algoritmo genético e do SGS. Um exemplo de aplicação é descrito pormenorizadamente em Magalhães-Mendes (2009).

8. TESTES DE DESEMPENHO

Este capítulo apresenta os resultados obtidos pelo algoritmo MM-GAV-FBI (*Genetic Algorithm - Random Key Variant with Forward Backward Improvement*). No estudo comparativo utiliza-se um conjunto de problemas teste descritos em Vaca (2005). Este conjunto consiste em 26 problemas. Cada problema é um projecto que tem até 80 actividades,

pode utilizar até 3 modos de execução e utilizar até 10 recursos do tipo renovável. Na Tabela 1 apresentam-se as características dos problemas teste.

Tabela 1 – Características dos problemas teste

Problema No.	No. Actividades	Complexidade ¹	Número de Recursos Renováveis	Número Médio de Modos por Actividade
1	24	30,91	3	2,41
2	21	12,15	4	2,14
3	15	17,27	4	2,33
4	12	7,33	2	1,75
5	16	8,98	3	2,18
6	40	24,20	5	2,87
7	55	18,13	4	1,87
8	73	69,82	10	1,54
9	27	24,12	3	2,63
10	80	37,24	2	1,43
11	51	35,87	2	1,69
12	7	13,77	4	2,00
13	7	5,85	3	1,71
14	43	14,61	4	1,95
15	13	10,65	4	1,46
16	7	8,10	2	1,62
17	9	13,08	3	2,77
18	12	13,65	3	1,57
19	13	9,62	3	1,14
20	13	7,29	4	1,26
21	16	17,42	2	1,68
22	16	10,8	6	1,47
23	27	19,39	3	1,77
24	30	12,07	6	1,63
25	27	24,12	3	2,77
26	11	5,78	3	2,72

Fonte: Vaca (1995)

Na resolução destes problemas o algoritmo genético usou a seguinte configuração:

- População = 5 x número de actividades do problema
- 1% de selecção elitista
- 0.1% de mutação
- 1 semente
- Critério de minimização: *makespan*
- Critério de paragem: 50 gerações.

O desempenho do algoritmo é baseado na comparação da melhor duração final

observada para cada problema com os resultados publicados nos estudos de Li e Wills (1992), Vaca (1995) e Badiru (1988). Na Tabela 2 pode-se observar que o algoritmo MM-GAV-FBI obtém os melhores resultados para o critério de avaliação (melhor duração que no presente trabalho significa a minimização do tempo de duração do projecto).

Tabela 2 – Comparação das durações para os problemas teste

Probl. No.	Duração Ótima Modo Único	Melhor Heurística ²	Duração Sem Restrição	Melhor Duração Modo Único	Duração Média Modo Único	Melhor Duração Múltiplos Modos	Duração Média Múltiplos Modos	MM-GAV-FBI
1	-	120	71	112	112	107	107,75	107
2	-	88	74	79	79	77	78,25	77
3	-	365	238	367	367	309	312,75	309
4	-	31	23	30	30	28	28,66	28
5	-	173	125	173	173	132	133,5	132
6	-	75	46	71	71	75	76,25	75
7	-	37	26	36	36	42	42,8	42
8		139	75	134	136,8	141	142,6	141
9	-	70	37	67	67	66	67,2	66
10	-	112	85	111	111	103	103,6	103
11	-	155	93	158	158,6	140	141	140
12	-	17	9	17	17	11	11	11
13	-	9	8	9	9	9	9	9
14		87	48	84	84	88	88	88
15	-	22	18	20	20	19	19	19
16	-	13	11	12	12	12	12	12
17	23	23	16	23	23	23	23,4	23
18	17	18	16	17	17	17	17,6	17
19	39	39	36	39	39	37	37	37
20	13	13	8	13	13,3	14	14	13
21	88	92	80	88	88	80	81	80
22	45	46	28	45	45,2	38	39	38
23	35	36	33	35	35,2	40	40	34
24	35	35	33	35	35	37	37,6	35
25	64	67	31	64	64,6	61	62	58
26	102	102	60	102	102	86	86,4	85

Fonte: Elaborado pelo autor

¹ Usando a fórmula proposta por Badiru (1988)

² Li e Wills (1992), Vaca (1995) e Badiru (1988)

O algoritmo foi implementado em Visual Basic 6.0 e os testes computacionais foram realizados num computador com sistema operativo Windows Vista e um processador Intel Core 2 Duo CPU T7250 @2.00 GHz. Os tempos médios de CPU para 50 gerações por tipo de problema não ultrapassam os 60 segundos.

9. CONCLUSÕES

Neste trabalho apresenta-se uma nova abordagem a ser utilizada no problema de programação de projectos com recursos limitados e em que cada actividade pode ter vários modos de execução. A representação cromossómica utilizada baseia-se em chaves aleatórias. A programação das actividades de cada projecto é realizada com recurso a um SGS baseado em prioridades definidas por um algoritmo genético. Este algoritmo genético é uma variante a uma abordagem inicialmente proposta por [1], no que respeita aos operadores genéticos.

Os testes de desempenho do algoritmo, realizados com base no conjunto de problemas propostos inicialmente por [18] e [31] e adaptados para múltiplos modos por [22], demonstram que o algoritmo MM-GAV-FBI tem um bom desempenho na obtenção de soluções com qualidade e em tempo útil.

REFERÊNCIAS

BADIRU, A. B.. Toward The Standardization of Performance Measures For Project Scheduling Heuristics, **IEEE Transaction on Engineering Management**. Vol. 35, n. 2, p. 80-89, 1988.

BLAZEWICZ, J.; LENSTRA, J. K.; RINNOOY KAN, A. H. G.. Scheduling subject to resource constraints: Classification and complexity, **Discrete Applied Mathematics**. Vol. 5, p. 11-24, 1983.

BOCTOR, F. F.. A new and efficient heuristic for scheduling projects with resource restrictions and multiple execution modes, **European Journal of Operational Research**. Vol. 90, p. 349-361, 1996.

_____. Heuristics for scheduling projects with resource restrictions and several resource-duration modes, **International Journal Production Research**. Vol. 31, No. 11, p. 2547-2558, 1993.

_____. Some Efficient Multi-Heuristic Procedures for Resource-Constrained Project Scheduling, **European Journal of Operational Research**. Vol. 49, p. 3-13, 1990.

DAMAK, N.; JARBOUI, B.; SIARRY, P.; LOUKIL, T.. Differential evolution for solving

multi-mode resource-constrained project scheduling problems, **Computers & Operations Research**. Vol. 36, p. 2653-2659, 2009.

DAVIS, W. E.. Project Scheduling Under Resource Constraints - Historical Review and Categorization of Procedures, **AIIE Transactions**. Vol. 5, n. 4, p. 147-163, 1973.

DAY, J. E.; HOTTENSTEIN, M. P.. **Review of Sequencing Research**. Naval Research Logistics Quarterly, Vol. 17, p. 11-40, 1970.

DE REYCK, B.; HERROELEN, W.. The multi-mode resource-constrained Project scheduling problem with generalized precedence relations, **European Journal of Operational Research**. Vol. 119, p. 538-556, 1999.

DREXEL, A.; GRUENEWALD, J.. Nonpreemptive Multi-Mode Resource-Constrained Project Scheduling, **IIE Transaction**. Vol. 25, n. 5, p. 74-81, 1993.

GOLDBERG, D. E.. **Genetic Algorithms in Search Optimization & Machine Learning**. Addison-Wesley, 1989.

HARTMANN, S.. Project Scheduling with Multiple Modes: A Genetic Algorithm, **Annals of Operations Research**. Vol. 102, p. 111-135, 2001.

HEILMANN, R.. Resource-constrained project scheduling: a heuristic for the multimode case, **OR Spektrum**. 23, p. 335-357, 2001.

HOLLAND, J. H. **Adaptation in Natural and Artificial Systems**. Ann Arbor: The University of Michigan Press, 1975.

JONES, R. A.. **Resource Scheduling: A Monte Carlo Approach to CPM**. Proceedings of CIB W-65, p. 422-427, 1992.

KLEIN, R.. Bidirectional planning: improving priority rule-based heuristics for scheduling resource constrained projects, **European Journal of Operational Research**. Vol. 127, p.619-638, 2000.

KOLISCH, R.; DREXL, A.. Local search for nonpreemptive multi-mode resource constrained project scheduling, **IIE Transactions**. Vol. 29, pp. 987-99, 1997.

KOLISCH, R.; HARTMANN, S.. Experimental investigation of heuristics for resource-constrained project scheduling: an update, **European Journal of Operational Research**. Vol.174 (1), p. 23-37, 2006.

LI, K. Y.; WILLS, R. J.. An Iterative Scheduling Technique for Resource-Constrained Project Scheduling, **European Journal of Operational Research**. Vol. 56, p. 370-379, 1992.

LOVA, A.; TORMOS, P.; Cervantes, M.; Barber, F.. An efficient hybrid genetic algorithm for scheduling projects with resource constraints and multiple execution modes, **International Journal of Production Economics**. Vol. 117, p. 302-316, 2009.

MACCARTHY, B. L.; LIU, J.. Addressing the Gap In Scheduling Research: A Review Of Optimization And Heuristic Methods In Production Scheduling, **International Journal of Production Research**. Vol. 31, n. 1, p. 59-79, 1993.

MAGALHÃES-MENDES, J.. **Project scheduling under multiple resources constraints using a genetic algorithm**. WSEAS TRANSACTIONS on BUSINESS and ECONOMICS, Issue 11, Volume 5, p. 487-496, 2008.

_____. Project Scheduling, **Progress in Management Engineering**. Chapter 6, Editors: Lucas P. Gragg e Jan M. Cassell, Nova Publishers, New York, USA, 2009.

MENDES, J. J. M., **Sistema de Apoio à Decisão para Planeamento de Sistemas de Produção do Tipo Projecto**. Tese de Doutoramento, Departamento de Engenharia Mecânica e Gestão Industrial, Faculdade de Engenharia da Universidade do Porto, Portugal, 2003.

MENDES, J. J. M.; GONÇALVES, J. F.; RESENDE, M. G. C.. A random key based genetic algorithm for the resource constrained project scheduling problem, **Computers & Operations Research**. Vol. 36, p. 92-109, 2009.

MORI, M.; TSENG, C.. A genetic algorithm for multi-mode resource constrained project scheduling problem. **European Journal of Operational Research**, Vol. 100, p. 134-141, 1997.

MOSELHI, O.; LORTERAPONG, P.. Near Optimal Solution for Resource-Constrained Scheduling Problems, **Construction Management and Economics**. Vol. 11, p. 293-303, 1993.

PATTERSON, J. H.; TALBOT, F. B.; SLOWINSKI, R.; WEGLARZ, J.. Computational Experience With A Backtracking Algorithm for Solving a General Class of Precedence and Resource-Constrained Scheduling Problems, **European Journal of Operational Research**. Vol. 49. p. 68-79, 1990.

RANJBAR, M.. Solving the resource-constrained project scheduling problem using filter-and-fan approach, **Applied Mathematics and Computation**. Vol. 201, p. 313–318, 2008.

SLOWINSKI, R.. Multiobjective Network Scheduling With Efficient Use of Renewable and Nonrenewable Resources, **European Journal of Operational Research**. Vol. 7, p. 265-273, 1981.

STORER, H. R.; WU, S. W.; VACCARI, R.. New Search Spaces For Sequencing Problems With Application to Job Shop Scheduling, **Management Science**. Vol. 38, n. 10, p. 1495-1509, 1992.

TALBOT, F. B.. Resource-Constrained Project Scheduling With Time-Resource Tradeoffs: the Nonpreemptive Case, **Management Science**. Vol. 28, n. 10, p. 1197-1210, 1982.

VACA, O. C. L.. **Um Algoritmo Evolutivo para a Programação de Projectos Multi-modos com Nivelamento de Recursos Limitados**. Tese de Doutorado, Universidade Federal de Santa Catarina, Florianópolis, Brasil, 1995.