

Estudo Comparativo da Utilização do Módulo de Fertilização InVitro em Algoritmos Genéticos para Resolução do Problema de Atraso Total Ponderado em Máquina Única*

Álefe C. Macedo¹
Hugo A. D. do Nascimento¹
Celso G. Camilo-Junior¹
Murilo B. Silva¹
Augusto C. S. Rocha¹

¹ Instituto de Informática – Universidade Federal de Goiás(UFG)
74.690-900 – Goiânia – GO – Brazil

alefemacedo@discente.ufg.br, hadna@ufg.br, celsocamilo@ufg.br,
murilo.silva@ifg.edu.br, augusto_cesar.cc@hotmail.com

Abstract. *This article presents a comparison that seeks to highlight the benefits of using the Genetic Algorithm with InVitro Fertilization in solving the single machine total weighted tardiness problem. Therefore, two other metaheuristics that were used to solve the problem are presented. In addition, we sought to define a set of experimental standards that would allow a fair comparison between the algorithms used. Finally, a collection of statistical data is presented in order to define the best method to solve the problem, as well as a description of the main contributions of the assisted fertilization module for Genetic Algorithms, and a set of suggestions for future work.*

Resumo. *Este artigo apresenta um comparativo que, busca salientar os benefícios da utilização do Algoritmo Genético com Fertilização InVitro na resolução do problema de atraso total ponderado em máquina única. Para tanto, são apresentadas outras duas meta-heurísticas que foram utilizadas na resolução do problema. Além disso, buscou-se definir um conjunto de passos experimentais que permitissem uma comparação justa entre os algoritmos utilizados. Por fim, é apresentada uma coleção de dados estatísticos de modo a definir qual o melhor método na resolução do problema, além de uma descrição das principais contribuições do módulo de fertilização assistida para os Algoritmos Genéticos, e um conjunto de sugestões para trabalhos futuros.*

1. Introdução

O Algoritmo Genético com Fertilização *InVitro* (AG+*InVitro*) caracteriza-se como uma variação do tradicional algoritmo genético que utiliza, paralelamente ao seu fluxo principal, um módulo que simula um processo de fertilização assistida. Esse módulo busca obter “super indivíduos” que sejam superiores ao melhor indivíduo gerado pelo algoritmo genético tradicional [Camilo-Junior and Yamanaka 2011].

Apesar desse algoritmo já ter sido aplicado a vários problemas de natureza combinatoria [Camilo-Junior and Yamanaka 2015], ele ainda não foi utilizado na resolução de

*Esta pesquisa foi apoiada pelo LaMCAD/UFG.

problemas do tipo *job-shop*. Deste modo, selecionou-se o Problema de Atraso Total Ponderado em Máquina Única (PATPMU), o qual é do tipo supracitado e que pertence à categoria dos problemas NP-difíceis, como objeto de estudo. Há uma gama de situações no mundo real em que o PATPMU ocorre, bem como muitas técnicas para resolvê-lo. Isso faz com que haja bons paralelos para apoiar uma análise comparativa [Liaw et al. 2003]. Na execução deste trabalho, foram escolhidas duas meta-heurísticas tradicionais para comparação com o AG+*InVitro* em termos de minimização de uma função objetivo.

A primeira meta-heurística utilizada é a Busca Tabu (BT), a qual aplica busca local [Glover and Laguna 1997]. A segunda, é uma implementação canônica dos algoritmos genéticos que utilizam busca populacional, de modo a otimizar o atraso total [Man et al. 1996].

Assim, esta pesquisa possui como objetivo geral investigar o desempenho de diferentes meta-heurísticas na resolução do PATPMU, bem como definir se a aplicação do módulo de fertilização *InVitro* em algoritmos genéticos é capaz de produzir resultados ainda melhores do que outras metodologias já existentes e que possuem histórico na resolução do problema. Para apoiar tal objetivo, foram levantados os seguintes objetivos específicos:

- Traçar um comparativo de desempenho entre o AG+*InVitro* e as outras heurísticas escolhidas.
- Identificar as contribuições advindas da utilização do módulo *InVitro* sobre o desempenho dos algoritmos genéticos na resolução do problema.

O restante desse trabalho está organizado da seguinte maneira: a Seção 2 traz uma sucinta revisão bibliográfica na área em que são apresentadas as meta-heurísticas e a base de dados utilizadas nesta pesquisa; já a Seção 3 constitui a metodologia adotada nesta pesquisa. Nela, são descritas as implementações das meta-heurísticas escolhidas para resolver o problema; em seguida, na Seção 4, são apresentadas as condições de teste escolhidas para os experimentos, bem como os resultados obtidos. Por fim, a Seção 5 trata de apresentar as principais conclusões levantadas acerca dos resultados obtidos, bem como de realizar o paralelo entre as três meta-heurísticas utilizadas de modo a determinar se o módulo de fertilização *InVitro* aplicado aos algoritmos genéticos é capaz de impulsionar os resultados destes.

2. Revisão Bibliográfica

Nesta seção, são apresentadas as teorias e componentes básicos do Algoritmo Genético com o módulo *InVitro* e o problema de atraso total ponderado em máquina única. Além disso, também é descrita a base de dados utilizada nos experimentos.

2.1. O Problema de Atraso Total Ponderado em Máquina Única

No PATPMU, são considerados n trabalhos a serem executados de maneira ininterrupta em uma única máquina com capacidade para processar um único trabalho por vez. Cada tarefa possui um tempo de execução positivo p_i , uma data de finalização d_i , tempo de conclusão C_i , peso w_i e se tornam disponíveis para processamento no instante zero. Assim, busca-se uma sequência de escalonamento não preemptiva para os processos de modo que o atraso ponderado T da execução dos n trabalhos seja o mínimo possível [Liaw et al. 2003, Liu et al. 2003, Rodrigues et al. 2008].

O atraso ponderado T_i da execução de uma tarefa i e o atraso total ponderado T resultante da execução dos n processos são dados, respectivamente, pelas Equações 1 e 2.

$$T_i = \max(0, C_i - d_i) \quad (1)$$

$$T = \sum_{i=1}^n w_i * T_i \quad (2)$$

2.2. Busca Tabu

A Busca Tabu pode ser definida como um procedimento adaptativo auxiliar que guia um algoritmo de busca local de maneira contínua na exploração de um espaço de busca. A Busca Tabu evita retornar a um ótimo local visitado previamente, o que reduz as chances de ficar presa em uma região do espaço de busca, ao mesmo tempo em que ajudar a encontrar um resultado ótimo ou próximo ao ótimo global.

De modo a encontrar a solução ótima, o algoritmo explora um conjunto de soluções X repetidas vezes, por meio de movimentos que partem de uma solução s para outra s' , localizada na vizinhança $N(s)$ de s , buscando minimizar uma função f [Gomes 2009, Glover and Laguna 1997].

O que permite ao processo de busca evitar ficar preso em ótimos locais é um mecanismo chamado lista Tabu. Essa estrutura de memória mantém, por um espaço de tempo, todos os movimentos de exploração que levam a locais já visitados previamente, que são então proibidos ou penalizados. O período de tempo pelo qual um movimento fica na lista Tabu é chamado prazo Tabu, e possui íntima ligação com o número de iterações do algoritmo e com o conjunto de movimento possíveis a partir da solução analisada [Gomes 2009].

Dentre os critérios de parada mais comuns para a BT, podemos citar:

- Uma solução ótima é encontrada;
- A vizinhança N de s na iteração $k + 1$ está vazia;
- Ou número de interações executadas desde que a melhor solução (s^*) mudou pela última vez é maior que o número máximo ($BTmax$) de interações especificado;

Outro componente muito importante para a Busca Tabu é a função A , a qual determina os critérios de aspiração. Estes critérios são procedimentos utilizados para decidir quando que movimentos, mesmo classificados como Tabu, poderão ser executados, ou seja, quando as restrições Tabu podem ser quebradas [Berger 2007].

2.3. Algoritmo Genético

Os algoritmos genéticos são heurísticas de busca adaptativa pertencentes ao conjunto dos algoritmos evolucionários, que produzem soluções ótimas ou quase ótimas para problemas de otimização ou de busca. Eles se baseiam no princípio da genética e da seleção natural, ou sobrevivência do mais apto, por meio do qual o indivíduo com as melhores características é capaz de se reproduzir e transmitir com sucesso seus genes para as gerações posteriores [Man et al. 1996].

Uma geração consiste em um conjunto de indivíduos que representam pontos no espaço de busca e, portanto, uma possível solução para o problema. Cada indivíduo, por sua vez, possui uma sequência de caracteres, inteiros, *floats* ou *bits*, análoga aos cromossomos, que o representa. Além disso, os indivíduos possuem uma pontuação de aptidão, ou *fitness*, obtida a partir de uma função objetivo, a qual irá determinar a capacidade de um indivíduo de ser escolhido para propagar seus genes para as próximas gerações [Man et al. 1996].

Existem quatro operações principais que são características de todo AG:

- **Operação de seleção:** busca selecionar os melhores indivíduos de acordo com as pontuações de aptidão e, assim, permitir que estes passem seus genes para as gerações posteriores.
- **Operação de cruzamento ou *crossover*:** após dois indivíduos serem selecionados pela operação de seleção, alguns pontos são escolhidos aleatoriamente dentre as posições dos alelos contidos nos cromossomos. Os alelos entre os pontos escolhidos são então trocados de modo a gerar um indivíduo novo com as características de ambos os pais. Essa operação pode ser realizada com um único ponto (*single-point crossover*) ou com dois ou mais pontos (*multi-point crossover*), e ocorre de acordo com certa taxa de probabilidade definida via configuração.
- **Operação de mutação:** responsável por inserir, de acordo com uma taxa probabilística, alelos em um descendente de modo a manter a diversidade na população e, assim, evitar convergências prematuras, bem como diversificar o espaço de busca.
- **Operação de sobrevivência:** define como será realizada a substituição dos indivíduos de uma população por seus descendentes, resultando na próxima geração. Pode ser utilizada uma estratégia onde todos os indivíduos da geração atual são substituídos por seus descendentes, ou pode-se optar por uma estratégia elitista na qual uma porcentagem dos melhores indivíduos são mantidos na população.

2.4. Algoritmo Genético com Fertilização *InVitro*

A principal diferença entre o AG com fertilização *InVitro* e o AG canônico está em sua capacidade de explorar características que provavelmente serão perdidas ao passar das gerações, já que muitos indivíduos são descartados durante o processo de escolha da população que representará a próxima geração. E, como alguns desses indivíduos possuem características que podem ser vantajosas para o processo de busca, recuperá-las pode ser impossível ou levar muito tempo [Camilo-Junior and Yamanaka 2011, Camilo-Junior and Yamanaka 2015]. Para tal, utiliza-se o módulo de fertilização *InVitro* que recebe uma porção da população atual e é executado em paralelo ao fluxo do AG tradicional. O fluxo é resumido a seguir:

1. Encontrar o melhor indivíduo da população e o nomeia como pai.
2. Receber um número N de indivíduos para serem recombinados com o pai, chamados mães.
3. Aplicar um operador que irá processar os indivíduos mães de modo a construir um conjunto N' .
4. Realizar o processo de divisão genética para particionar o cromossomo pai.
5. Recombinar o indivíduo pai e o conjunto N' de mães de acordo com o resultado da divisão genética.

6. Caso os indivíduos gerados sejam bons, os inserir na população.

Existem três componentes principais no módulo de fertilização *InVitro*:

- **Divisão genética:** o processo de divisão genética busca separar os cromossomos dos indivíduos em grupos de genes ou gene a gene, antes do processo de recombinação. O conhecimento do projetista do algoritmo acerca do problema pode ajudar a definir a melhor maneira de dividir os cromossomos e assim auxiliar na otimização.
- **Operadores:** os operadores buscam apoiar o AG na exploração do espaço de busca. O operador de Recombinação Assistida (Assisted Recombination operator - AR) procura recombinar os indivíduos inalterados da população. Já os operadores de Recombinação Assistida Exploratória (Exploratory Assisted Recombination operators – EAR) efetuam alterações em $N/2$ indivíduos mãe antes de recombiná-los com o pai, e são nomeados como EAR-T, EAR-P e EAR-N. Dentre os operadores EAR, o EAR-N altera a população de mães substituindo metade dela por novos indivíduos.
- **Recombinação:** ocorre após as etapas de divisão genética e da construção da população N' pelos operadores. Nesta etapa, os cromossomos são representados como vetores e cada posição destes contém um grupo de alelos, ou conjunto de genes, obtido pela divisão genética. O processo de recombinação gera um indivíduo para cada conjunto de genes do pai, com todas as mães do conjunto N' . Ou seja, se os cromossomos foram divididos em três partes, serão gerados três indivíduos filhos para cada mãe. Os indivíduos de cada conjunto de filhos são então agrupados, e o melhor tem a chance de concorrer para se tornar o super indivíduo que, possivelmente, substituirá o pai na próxima iteração [Camilo-Junior and Yamanaka 2011].

3. Método

Nesta pesquisa, tanto as soluções avaliadas pela Busca Tabu quanto os cromossomos dos indivíduos utilizados nos Algoritmos Genéticos, com e sem o módulo *InVitro*, são representações da sequência de execução dos processos a serem escalonados. Além disso, nos algoritmos genéticos, o atraso total ponderado representa o *fitness* de um indivíduo.

Devido à natureza dos dados a serem processados, foram necessárias algumas alterações nas implementações do Algoritmo Genético canônico e do módulo *InVitro* definidas na literatura. Essas alterações ocorreram nos operadores de cruzamento e mutação, no operador de recombinação do *InVitro* e na função objetivo.

O operador de mutação foi implementado de modo que um alelo não seja substituído, e sim deslocado de maneira aleatória para outra posição do cromossomo, sem alterar a sequência dos outros identificadores. Deste modo mantém-se a unicidade dos identificadores contidos nos cromossomos, uma vez que em uma sequência de execução todos os processos são executados até sua completude não havendo, portanto, a execução repetida de um mesmo processo.

Assim como na operação de mutação, a operação de cruzamento também necessitou de cuidados durante a manipulação dos alelos que constituem os cromossomos dos indivíduos. O operador de cruzamento implementado utiliza a estratégia *two-point crossover two children (Version I) (TPC2CV1)*, definida por Ferrolho e Crisóstomo (2007).

Nesta, dois pontos (p_1 e p_2) aleatórios são definidos de modo a realizar a operação e retornar dois descendentes, como mostrado na Figura 1a. Para dois indivíduos (i_1 e i_2) com cromossomos de tamanho n , tem-se que todos os alelos a partir da posição inicial 0 até p_1 e de p_2 até a posição final $n - 1$, do indivíduo i_1 , são atribuídos ao primeiro filho nas posições equivalentes. As posições restantes, partindo de $p_1 + 1$ até $p_2 - 1$, serão preenchidas pelos identificadores que não foram recebidos do indivíduo i_1 , na sequência definida pelo indivíduo i_2 . O processo é então repetido para o segundo filho, todavia, utilizando primeiramente a sequência dos alelos do indivíduo i_2 .

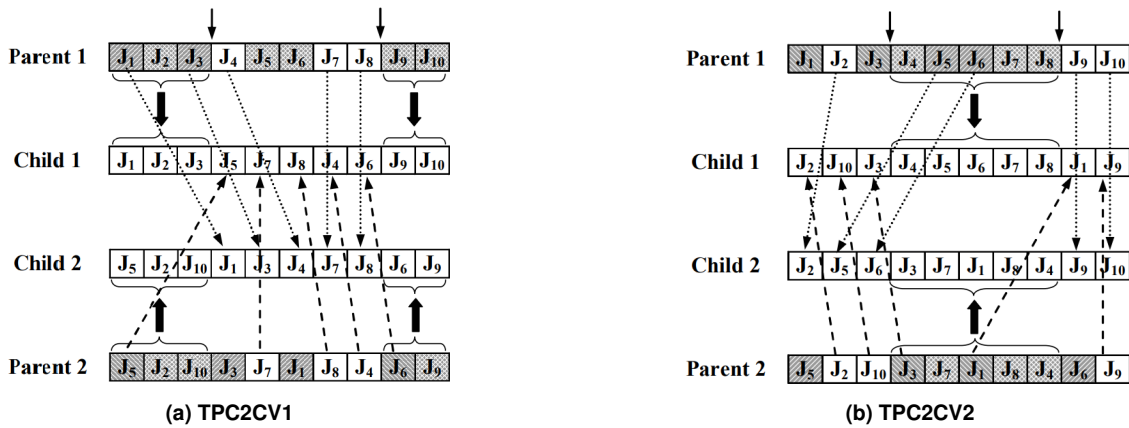


Figura 1. Operadores de Cruzamento

Fonte: [Ferrolho and Crisostomo 2007]

Como definido pelo problema abordado, a função objetivo F deve retornar o atraso total ponderado definido pela sequência de processos/*jobs* contida no cromossomo, para os algoritmos genéticos e o módulo *InVitro*, ou na solução a ser avaliada pela Busca Tabu.

Para tanto, a função objetivo percorre todos os identificadores de *jobs* e de acordo com o peso (w_i), tempo de execução (p_i) e tempo de finalização (d_i) definidos para cada um, calcula o atraso ponderado do processo em questão e então soma-o ao atraso total como definido pela Equação 2.

Devido à característica de que, quanto menor o valor do atraso para um indivíduo, melhor é a qualidade dos seus genes, o *fitness* também é tratado como custo.

O módulo *InVitro* sofreu alterações apenas em seu processo de recombinação e mutação. No processo de mutação utilizou-se a mesma estratégia adotada no operador de mutação do Algoritmo Genético canônico, com exceção de que este pode ser executado em apenas uma parte do cromossomo, como requerido pelo operador EAR-P. Já no processo de recombinação aplicou-se a estratégia *two-point crossover two children (Version II)* (TPC2CV2) novamente definida por [Ferrolho and Crisostomo 2007] (Figura 1b). Todavia, os pontos onde a recombinação será realizada não são mais aleatórios e sim definidos pelo processo de divisão genética do próprio *InVitro*, e somente o primeiro filho é retornado.

Ao analisar o processo de busca definido pela arquitetura do AG+*InVitro* é fácil notar que este possibilita uma maior exploração do espaço de busca, quando comparado ao AG canônico e a Busca Tabu, devido ao maior número de avaliações que realiza por geração. Assim, visando um comparativo mais justo entre as meta-heurísticas foram de-

finidos alguns critérios de parada para o AG+*InVitro* e AG canônico que não existem na literatura original.

Para a Busca Tabu o número de iterações sem melhora foi definido em 100 e nenhum outro critério de parada foi definido. O AG canônico recebeu como critério de parada um número máximo de gerações sem melhora fixado em 50, além de ter o número de avaliações da função objetivo fixado em 65.000, ou seja, este pode avaliar no máximo um total de 65.000 indivíduos, excluindo as mutações. Já o AG+*InVitro* recebeu os mesmos critérios de parada do AG canônico referentes ao número de gerações sem melhora e de avaliações, todavia, de modo a impedir o gasto de avaliações o módulo *InVitro* foi configurado de modo a ser executado apenas a cada 10 gerações.

De modo a verificar como a diversidade de indivíduos decresce ao passar das gerações nos algoritmos genéticos, foi implementada uma função que calcula a distância entre os indivíduos de uma população P de tamanho N . Para cada par de indivíduos (i_a, i_b) , com $1 \leq a, b \leq N$, a distância $D_{a,b}$ é dada pelo valor absoluto da subtração entre os *fitness* de i_a e i_b retornados pela função objetivo F , como mostrado na Equação 3. A *distância dentro da população* D_p , Equação 4, é então obtida a partir da média aritmética das distâncias $D_{a,b}$, com $C_{N,2}$ a quantidade de pares únicos dos N indivíduos da população.

$$D_{a,b} = \text{abs}(F(i_a) - F(i_b)) \quad (3)$$

$$D_p = \frac{1}{C_{N,2}} \sum_{1 \leq a < b \leq N} D_{a,b} \quad (4)$$

Em todos os experimentos realizados foi utilizado uma única base de dados disponibilizada especificamente para o problema de atraso total ponderado em máquina única, e que apesar de conter dados fictícios mostra-se como uma boa representação do problema no contexto do mundo real.

Esta contém instâncias geradas de maneira aleatória, de tamanho $n = 40$, $n = 50$ e $n = 100$ *jobs*, constituindo as três complexidades construídas para o problema. Cada processo ou *job* possui um identificador j ($j = 1, \dots, n$), um tempo de processamento inteiro p_j , obtido a partir da distribuição uniforme $[1, 100]$, um peso w_j gerado a partir da distribuição uniforme $[1, 10]$. Por fim, cada processo possui também uma data de finalização d_j obtida a partir da distribuição uniforme $[P(1 - TF - RDD/2), P(1 + TF + RDD/2)]$, onde o *relative range of due dates* (*RDD*) é dado por $RDD = (0.2, 0.4, 0.6, 0.8, 1.0)$, o *average tardiness fator* $TF = (0.2, 0.4, 0.6, 0.8, 1.0)$ e $P = \sum_{j=1}^n p_j$. Para cada um dos 25 pares de RDD e TF cinco instâncias foram geradas, resultando em um total de 125 instâncias por complexidade [Congram et al. 2002].

4. Avaliação

Visando obter um comparativo justo e confiável entre as meta-heurísticas foi construído um experimento onde as mesmas condições de teste foram mantidas para cada método. Devido aos processos serem estocásticos, o experimento foi montado de modo que todos os métodos fossem aplicados um total de 100 vezes sobre um conjunto de 25 instâncias de complexidades $n = 40$ e $n = 50$ do banco de dados. A partir das 100 execuções, foram

obtidos os valores das métricas de média, mediana, variância e desvio padrão para cada instância que, por sua vez, foram utilizados para a realização de um teste estatístico de normalidade de modo a estabelecer os níveis de confiança dos dados.

A partir do teste estatístico de normalidade foi então definido se seria aplicado o teste paramétrico *T-Student* ou o teste não paramétrico de Mann-Whitney. Utilizando estes testes, buscou-se identificar qual método teve o melhor desempenho na resolução do problema em cada uma das instâncias e, ao final, qual o método com o melhor desempenho geral.

Além disto, buscando identificar quais foram as contribuições do módulo *InVitro* sobre o desempenho dos algoritmos genéticos, foram realizados experimentos com duas instâncias de complexidade de 40 e 50 processos respectivamente. A configuração dos experimentos foi muito parecida à descrita anteriormente: para cada instância, o AG canônico sem o módulo *InVitro* e o AG com o módulo *InVitro* foram executados 100 vezes, sendo que, em cada execução, partiu-se da mesma solução inicial; a diferença foi que, para cada execução, foram realizadas um total de 70000 avaliações e também foi construído um histórico dos melhores indivíduos e das distâncias entre os indivíduos das populações.

Por meio do histórico de *fitness* dos melhores indivíduos, foi produzido um gráfico contendo os valores de mediana, mínimo e máximo das 100 execuções do AG canônico e do AG+*InVitro* por geração. Deste modo, foi possível comparar de maneira visual a evolução de cada um dos modelos ao passar das gerações. Também foi plotada uma reta com o resultado ótimo definido pela base de dados para cada uma das instâncias.

Um outro gráfico foi construído utilizando o histórico das distâncias entre os indivíduos das populações testadas, comparando os AGs com e sem o módulo *InVitro*. Neste, também foram apresentados os valores de mediana, mínimo e máximo das 100 execuções, para cada geração. Essa representação tem como objetivo mostrar como a diversidade da população de indivíduos decresce ao passar das gerações, e como o módulo *InVitro* pode interferir nesse quesito.

Em todos os experimentos com o módulo *InVitro*, foram utilizadas as proporções 0.2, 0.5 e 0.3 para a divisão genética, o número de indivíduos selecionados para serem mães foi configurado em 30, as recombinações foram limitadas para um máximo de 100 e o operador escolhido para processar os indivíduos mãe foi o EAR-N. Além disso, as taxas de mutação e cruzamento foram definidas em 10% e 90%, respectivamente, e foi utilizada uma estratégia de torneio do tipo *Top-5* na seleção dos indivíduos para cruzamento, inclusive para o AG canônico.

4.1. Resultados obtidos

A Tabela 1 e a Tabela 2 apresentam os valores métricos obtidos nas 100 execuções de cada modelo, bem como o valor P do Teste U de Mann-Whitney e a classificação entre os modelos. As métricas escolhidas são média, mediana, desvio padrão (StDv), mínimo (min) e máximo (max). Com essas métricas, em especial o valor médio, é possível determinar qual método tem o melhor desempenho para cada instância. O valor P determina se é possível comparar dois modelos para uma determinada instância. Se o valor P for menor que 0,05 (5%) então a comparação é possível e os valores médios definem quem é o melhor entre os dois modelos comparados.

Tabela 1. Valores Estatísticos para Instâncias com 40 Jobs

#inst	BT	AG					AG+InV.					P-Values			Ranking
		Média	StDv	Mediana	Min	Max	Média	StDv	Mediana	Min	Max	BT x AG	BT x InV.	AG x InV.	
1	956	952.78	21.79	956	913	1012	955.33	29.35	956	913	1094	2.24e-4	5.35e-3	0.42	AG>BT, Inv.>BT
2	1225	1277.52	60.97	1263	1225	1386	1293.69	71.05	1267	1225	1513	1.72e-24	2.06e-25	0.14	BT>AG, BT>Inv.
3	951	580.86	66.33	573	537	951	588.78	84.22	573	537	951	3.75e-41	1.95e-39	0.24	AG>BT, Inv.>BT
4	2121	2114.35	59.15	2094	2094	2379	2121.71	62.21	2094	2094	2376	1.26e-20	3.02e-15	0.10	AG>BT, BT>Inv.
5	990	1002.08	45.13	990	990	1244	1001.56	40.04	990	990	1149	3.65e-3	2.02e-3	0.42	BT>AG, BT>Inv.
6	7048	6960.39	16.99	6955	6955	7053	6957.86	8.94	6955	6955	7024	5.31e-39	9.00e-42	0.40	AG>BT, Inv.>BT
7	6324	6390.12	103.75	6324	6324	6580	6372.79	93.90	6324	6324	6607	1.20e-14	2.55e-11	0.07	BT>AG, BT>Inv.
8	7146	6871.51	12.28	6865	6865	6910	6869.95	10.59	6865	6865	6901	3.79e-41	1.13e-41	0.18	AG>BT, Inv.>BT
9	16281	16347.76	118.95	16320	16225	16700	16338.75	113.74	16320	16225	16671	4.48e-3	9.06e-3	0.32	BT>AG, BT>Inv.
10	10109	9791.99	105.50	9771	9737	10143	9783.64	86.72	9771	9737	10127	9.72e-33	1.74e-35	0.28	AG>BT, Inv.>BT
11	17465	17542.87	74.14	17562	17465	17876	17559.52	125.40	17562	17465	18065	1.10e-28	6.50e-25	0.41	BT>AG, BT>Inv.
12	19312	19375.34	74.39	19329	19312	19706	19376.54	70.64	19335	19312	19714	8.16e-26	3.76e-32	0.26	BT>AG, BT>Inv.
13	29287	29424.69	108.52	29431	29256	29830	29405.71	94.21	29431	29256	29857	1.52e-28	1.28e-30	0.07	BT>AG, BT>Inv.
14	14529	14447.23	34.22	14434	14377	14574	14439.01	33.24	14432	14377	14517	1.62e-36	1.77e-39	5.09e-2	AG>BT, Inv.>BT
15	26914	27113.75	201.09	27139	26914	27944	27123.42	252.53	27139	26914	27934	2.49e-35	1.18e-32	0.15	BT>AG, BT>Inv.
16	72564	72658.92	234.38	72585	72362	73367	72567.84	219.30	72502	72322	73366	0.15	4.52e-3	3.18e-4	BT>Inv, Inv.>AG
17	78747	78797.35	201.65	78691	78623	79596	78792.32	192.32	78690	78623	79246	2.04e-3	3.00e-3	0.19	BT>AG, BT>Inv.
18	74373	74606.67	225.98	74536	74330	75709	74536.65	172.75	74491	74310	75333	3.40e-32	2.16e-22	9.84e-3	BT>AG, BT>Inv., Inv.>AG
19	77432	77544.76	175.97	77565	77146	78057	77479.35	163.24	77485	77125	77958	8.97e-18	7.92e-09	1.23e-3	BT>AG, BT>Inv., Inv.>AG
20	63368	63880.52	421.20	63904	63273	65262	63751.98	482.96	63564	63240	65826	2.68e-28	2.34e-15	1.00e-3	BT>AG, BT>Inv., Inv.>AG
21	77774	77892.01	90.69	77872	77784	78281	77833.07	55.93	77820	77774	78061	2.82e-39	1.51e-37	7.29e-10	BT>AG, BT>Inv., Inv.>AG
22	100484	100635.75	113.21	100609	100491	101021	100554.7	59.87	100540	100487	100821	2.82e-39	2.80e-39	5.37e-10	BT>AG, BT>Inv.
23	135618	135778.5	119.45	135749	135632	136355	135697.1	74.20	135678	135618	136034	2.82e-39	4.06e-38	2.37e-11	BT>AG, BT>Inv., Inv.>AG
24	119947	120212.1	162.62	120178	119982	120711	120105.42	139.74	120052	119948	120603	2.82e-39	2.82e-39	2.77e-08	BT>AG, BT>Inv., Inv.>AG
25	128747	128948.14	156.90	128919	128764	129684	128850.28	98.61	128818	128753	129192	2.82e-39	2.81e-39	2.54e-09	BT>AG, BT>Inv., Inv.>AG

Tabela 2. Valores Estatísticos para Instâncias com 50 Jobs

#inst	BT	AG					AG+Inv.					P-Values			Ranking
		Média	StDv	Mediana	Min	Max	Média	StDv	Mediana	Min	Max	BT x AG	BT x Inv.	AG x Inv.	
1	2184	2178.41	76.14	2134	2134	2346	2160.54	58.85	2134	2134	2356	2.78e-10	2.50e-19	0.08	AG>BT, Inv.>BT
2	1998	2004.31	7.77	1998	1996	2044	2004.3	6.47	2008	1996	2011	2.54e-09	4.21e-13	0.39	BT>AG, BT>Inv.
3	2583	2606.15	56.99	2583	2583	2948	2609.69	68.32	2583	2583	2863	5.24e-08	2.54e-06	0.20	BT>AG, BT>Inv.
4	2691	2696.56	29.11	2691	2691	2915	2703.85	62.93	2691	2691	3112	6.64e-3	3.66e-3	0.38	BT>AG, BT>Inv.
5	1604	1576.32	67.11	1604	1518	1996	1589.66	77.96	1604	1518	1996	2.85e-11	1.28e-05	0.06	AG>BT, Inv.>BT
6	26415	26459.81	134.47	26430	26281	26922	26425.42	120.81	26427	26276	26876	1.29e-06	8.63e-4	0.04	BT>AG, BT>Inv., Inv.>AG
7	11403	11487.2	151.63	11409	11403	12300	11542.5	185.97	11415	11403	12168	3.17e-27	1.26e-28	0.04	BT>AG, BT>Inv., AG>Inv.
8	8760	8649.28	115.79	8634	8499	9001	8643.07	125.86	8630	8499	9132	6.20e-22	1.41e-22	0.25	AG>BT, Inv.>BT
9	9923	9924.38	16.78	9918	9884	9993	9929.39	20.88	9923	9884	10051	1.51e-09	2.53e-3	5.16e-2	BT>AG, BT>Inv.
10	10945	10699.42	80.76	10668	10655	10999	10712.19	102.51	10660	10655	11085	7.09e-38	2.70e-32	0.38	AG>BT, Inv.>BT
11	44028	44022.03	251.87	44006	43526	44915	43880.86	218.29	43906	43507	44799	5.87e-2	3.39e-18	9.99e-07	Inv.>BT, Inv.>AG
12	36648	36865.5	218.41	36825	36471	37738	36800.34	192.49	36765	36470	37750	4.66e-27	3.35e-19	1.39e-3	BT>AG, BT>Inv., Inv.>AG
13	45967	46079.19	344.38	46091	45426	47405	45827.62	267.89	45777	45391	46449	4.49e-05	8.81e-08	4.52e-08	BT>AG, Inv.>BT, Inv.>AG
14	52039	52202.31	140.22	52160	51936	52710	52140.34	111.87	52119	51849	52626	1.49e-33	2.68e-28	2.89e-4	BT>AG, BT>Inv., Inv.>AG
15	39068	39227.67	213.87	39224	38939	39994	39121.28	152.56	39074	38934	39470	4.59e-09	0.26	6.03e-05	BT>AG, Inv.>AG
16	87902	89517.74	753.97	89347	88476	92220	88658.74	453.65	88542	88009	90155	2.82e-39	2.82e-39	4.33e-19	BT>AG, BT>Inv., Inv.>AG
17	84375	85433.35	595.47	85213	84697	87375	84801.56	232.13	84766	84444	85789	2.82e-39	2.82e-39	1.49e-21	BT>AG, BT>Inv., Inv.>AG
18	104795	106803.15	718.97	106695	105499	109850	106032.12	589.19	105849	105105	108194	2.82e-39	2.82e-39	1.61e-14	BT>AG, BT>Inv., Inv.>AG
19	89474	90314.32	547.44	90253	89431	93453	89875.62	329.45	89835	89336	90981	8.42e-38	1.50e-33	3.79e-12	BT>AG, BT>Inv., Inv.>AG
20	72356	73191.52	330.93	73154	72555	74343	72857.79	261.42	72828	72372	73584	2.82e-39	2.82e-39	4.64e-13	BT>AG, BT>Inv., Inv.>AG
21	214566	216109.86	609.28	216049	215038	218339	215248.23	322.80	215164	214704	216740	2.82e-39	2.82e-39	7.73e-24	BT>AG, BT>Inv., Inv.>AG
22	150800	151890.7	474.69	151749	151238	153513	151241.62	192.63	151207	150926	151988	2.82e-39	2.82e-39	7.38e-27	BT>AG, BT>Inv., Inv.>AG
23	224025	225838.57	745.42	225708	224607	228441	224790.77	397.45	224694	224253	226144	2.82e-39	2.82e-39	1.58e-24	BT>AG, BT>Inv., Inv.>AG
24	116015	117411.81	531.58	117280	116622	120013	116669.18	275.55	116610	116282	117738	2.82e-39	2.82e-39	6.84e-26	BT>AG, BT>Inv., Inv.>AG
25	240179	241976.23	860.45	241716	240689	246004	240837.88	354.02	240734	240387	242202	2.82e-39	2.82e-39	1.07e-27	BT>AG, BT>Inv., Inv.>AG

Analisando os dados da Tabela 1 e da Tabela 2, percebe-se que a Busca Tabu obteve o melhor desempenho entre os três algoritmos testados, tanto para as instâncias com 40 jobs, quanto para as instâncias com 50 jobs. Na Tabela 1 existem 24 instâncias onde o comparativo entre a Busca Tabu e o AG canônico é possível, sendo impossível apenas na instância #15. Dessas 24 instâncias, a Busca Tabu obteve um desempenho superior em 17, tendo, então, um percentual de sucesso de 70,83%. Para a Busca Tabu e o AG+InVitro, foi possível compará-los em todas as 25 instâncias. Delas, a Busca Tabu obteve o melhor desempenho em 19 e o AG+InVitro em 6, marcando assim um percentual de sucesso de 76% para a Busca Tabu sobre o AG+InVitro.

O GA+InVitro e o AG canônico foram comparáveis em apenas 9 das 25 instâncias com 40 jobs. Em todas essas instâncias, o AG+InVitro obteve um desempenho superior ao do AG canônico, atingindo um percentual de sucesso de 100%.

Para as instâncias de 50 processos listadas na Tabela 2, a comparação entre a Busca Tabu e o AG canônico foi possível em 24 instâncias (na instância #11, o comparativo não foi realizável). Das 24 instâncias, a Busca Tabu foi vitoriosa em 20, obtendo, assim, um

percentual de sucesso de 83%. Já o AG+*InVitro* foi comparável à Busca Tabu em 24 instâncias (não foi possível comparar na instância #15), sendo a Busca Tabu o melhor algoritmo em 18 delas, atingindo um percentual de sucesso de 75%.

Por outro lado, o AG canônico foi comparável ao AG+*InVitro* em 16 instâncias com 50 processos. Dessas instâncias o AG+*InVitro* foi melhor em 15, marcando um percentual de sucesso de 93,75% sobre o AG canônico.

Apesar de o AG+*InVitro* não apresentar um desempenho elevado quando comparado à Busca Tabu, tendo sido superior em apenas 6 instâncias em ambos experimentos, ainda assim obteve um desempenho satisfatório quando comparado ao AG canônico, pois o AG+*InVitro* atingiu percentuais de sucesso de 100% e 93,75% sobre AG canônico para instâncias com 40 e 50 processos respectivamente.

Além disso, os resultados da Busca Tabu são os mesmos em todas as execuções, portanto não há variação. Se os valores mínimos do AG+*InVitro* forem utilizados no comparativo, é possível que o AG+*InVitro* tenha sucesso em mais instâncias do que a própria Busca Tabu.

No geral, para a resolução do problema de atraso total ponderado em máquina única, o algoritmo de Busca Tabu apresentou um desempenho médio melhor do que o AG+*InVitro* e o AG canônico, sendo, então, o método mais indicado para resolver o problema. Em seguida temos o AG+*InVitro* como o segundo algoritmo mais recomendado. Já o AG canônico se mostrou o método menos recomendado para a resolução deste problema, uma vez que obteve um desempenho bem inferior ao dos outros algoritmos.

A partir do processamento de duas instâncias com complexidades de 40 e 50 processos e do histórico dos melhores indivíduos avaliados ao longo das gerações, foram construídos os gráficos das Figuras 2 a 5 para comparar o desempenho da implementação do AG com e sem o módulo *InVitro* ativo. Como o número de avaliações foi limitado (70.000) e o AG+*InVitro* realiza mais avaliações da função objetivo, comparado ao AG canônico, seu número de gerações realizadas é menor.

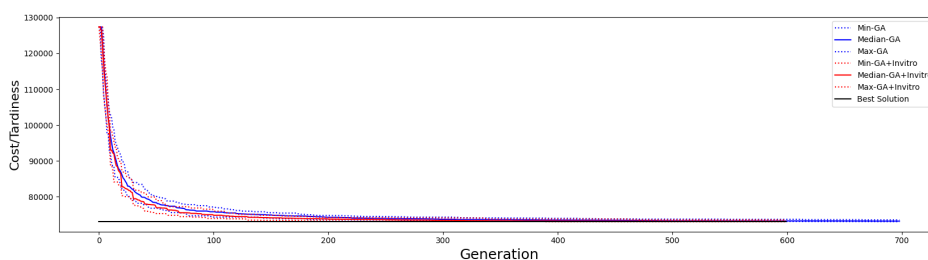


Figura 2. Evolução dos algoritmos genéticos ao longo das gerações (Instância 124, com 40 processos)

Analisando os gráficos da Figura 2 e Figura 3, nota-se que o AG+*InVitro* possui uma queda mais acentuada em suas retas de mediana, mínimo e máximo quando comparado ao AG canônico. Isto demonstra uma melhora mais rápida no *fitness* do melhor indivíduo, conseguindo assim se aproximar em poucas gerações da reta que representa a solução ótima. Além disso, os valores denotados pela reta de mínimo do AG+*InVitro* são inferiores ao da reta de mínimo do AG canônico, demonstrando um desempenho superior

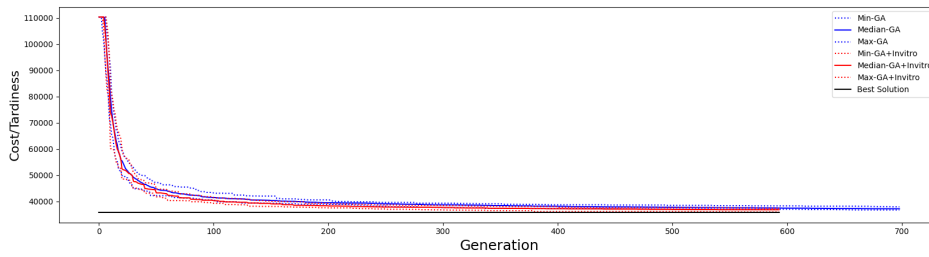


Figura 3. Evolução dos algoritmos genéticos ao longo das gerações (Instância 115, com 50 processos)

na minimização da função objetivo para um mesmo número de gerações nas duas instâncias analisadas. Mesmo que a reta de mínimo do AG com o módulo *InVitro* desativado apresente alguns pontos de mínimo inferiores ao AG com *InVitro*, a performance geral deste último ainda é superior.

Os gráficos da Figura 4 e Figura 5 mostram a perda de diversidade nas populações de indivíduos ao longo das gerações, além de mostrar como o módulo de fertilização ajuda a mitigar os efeitos nocivos decorrentes dessa perda.

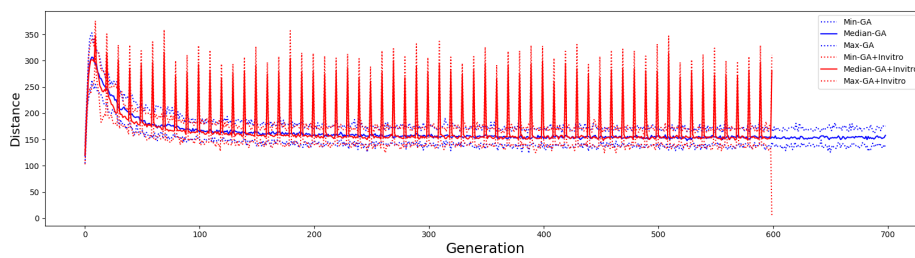


Figura 4. Perca da diversidade na população de indivíduos (Instância 124, com 40 processos)

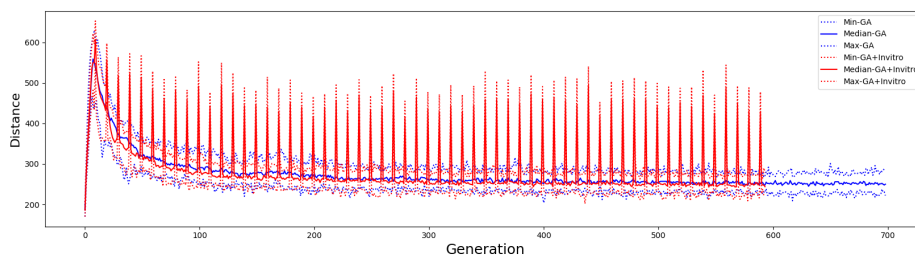


Figura 5. Perca da diversidade na população de indivíduos (Instância 115, com 50 processos)

Os picos acentuados no gráfico da Figura 4 e Figura 5 representam as inserções de super-indivíduos gerados pelo módulo *InVitro* na população. Isso retarda a queda na diversidade populacional, permitindo que o AG explore melhor o espaço de busca e evite mais facilmente os mínimos locais. Como o AG canônico não possui um mecanismo que impeça que a diversidade populacional caia vertiginosamente, é mais provável que caia nos pontos mínimos locais, dificultando assim a busca do ótimo global.

5. Conclusões

Nesta pesquisa, vimos que os Algoritmos Genéticos com Fertilização *InVitro* fornecem contribuições significativas para a resolução do problema do atraso ponderado, conforme mostram os dados estatísticos obtidos a partir do processamento dos resultados das 100 execuções de Busca Tabu, AG canônico e AG+*InVitro*.

Partindo de uma análise temporal, nota-se que o AG+*InVitro* tem mediana, média e valores máximos inferiores aos do AG canônico pelo maior tempo possível. O AG+*InVitro* atinge, além disso, um valor mínimo com menos gerações, conseguindo obter valores de custo mais baixos para quase todas as instâncias em que foi possível realizar uma comparação entre os modelos.

Por fim, espera-se que, em trabalhos futuros, seja processado um número maior de instâncias com o intuito de aumentar os dados disponíveis para análise. Espera-se, também, testar novos parâmetros de configuração para o AG+*InVitro*, como taxas de mutação e cruzamento, visando melhorar seu desempenho em relação à Busca Tabu.

Referências

- Berger, M. (2007). Aplicação da Busca Tabu no Gerenciamento de Pilhas em Pátios de Armazenamento de Placas de Aço. Dissertação, Universidade Federal do Espírito Santo.
- Camilo-Junior, C. and Yamanaka, K. (2015). In vitro fertilization genetic algorithm applied to multidimensional 0-1 knapsack problem. *Proceedings of the 2011 International Conference on Artificial Intelligence, ICAI 2011*, 2.
- Camilo-Junior, C. G. and Yamanaka, K. (2011). In vitro fertilization genetic algorithm. In Kita, E., editor, *Evolutionary Algorithms*, chapter 4. IntechOpen, Rijeka.
- Congram, R. K., Potts, C. N., and van de Velde, S. L. (2002). An Iterated Dynasearch Algorithm for the Single-Machine Total Weighted Tardiness Scheduling Problem. *INFORMS Journal on Computing*, 14(1):52–67. Publisher: INFORMS.
- Ferrolho, A. and Crisostomo, M. (2007). Single machine total weighted tardiness problem with genetic algorithms. In *2007 IEEE/ACS International Conference on Computer Systems and Applications*, pages 1–8.
- Glover, F. W. and Laguna, M. (1997). Tabu search. In *Handbook of Heuristics*.
- Gomes, A. (2009). Uma Introdução à Busca Tabu. page 27.
- Liaw, C.-F., Lin, Y.-K., Cheng, C.-Y., and Chen, M. (2003). Scheduling unrelated parallel machines to minimize total weighted tardiness. *Computers & Operations Research*, 30(12):1777–1789.
- Liu, N., Abdelrahman, M., and Ramaswamy, S. (2003). A genetic algorithm for the single machine total weighted tardiness problem. In *Proceedings of the 35th Southeastern Symposium on System Theory, 2003.*, pages 34–38.
- Man, K., Tang, K., and Kwong, S. (1996). Genetic algorithms: Concepts and applications. *IEEE Transactions on Industrial Electronics*, 43(5):519–534.
- Rodrigues, R., Pessoa, A., Uchoa, E., and de Aragao, M. P. (2008). Heuristic Algorithm for the Parallel Machine Total Weighted Tardiness Scheduling Problem. 8(10):12.