

UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

ÉWERTON CARLOS DE ARAÚJO ASSIS

Heurísticas e metaheurísticas aplicadas ao Problema de Escalonamento Job-Shop

**Um algoritmo evolucionário híbrido baseado na
fertilização in vitro para solucionar problemas de
escalonamento job-shop**

Goiânia
2011

UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

**AUTORIZAÇÃO PARA PUBLICAÇÃO DE TRABALHO DE
CONCLUSÃO DE CURSO EM FORMATO ELETRÔNICO**

Na qualidade de titular dos direitos de autor, **AUTORIZO** o Instituto de Informática da Universidade Federal de Goiás – UFG a reproduzir, inclusive em outro formato ou mídia e através de armazenamento permanente ou temporário, bem como a publicar na rede mundial de computadores (*Internet*) e na biblioteca virtual da UFG, entendendo-se os termos “reproduzir” e “publicar” conforme definições dos incisos VI e I, respectivamente, do artigo 5º da Lei nº 9610/98 de 10/02/1998, a obra abaixo especificada, sem que me seja devido pagamento a título de direitos autorais, desde que a reprodução e/ou publicação tenham a finalidade exclusiva de uso por quem a consulta, e a título de divulgação da produção acadêmica gerada pela Universidade, a partir desta data.

Título: Heurísticas e metaheurísticas aplicadas ao Problema de Escalonamento Job-Shop – Um algoritmo evolucionário híbrido baseado na fertilização in vitro para solucionar problemas de escalonamento job-shop

Autor(a): Éwerton Carlos de Araújo Assis

Goiânia, 19 de Dezembro de 2011.

Éwerton Carlos de Araújo Assis – Autor

Prof. Celso Gonçalves Camilo Júnior – Orientador

ÉWERTON CARLOS DE ARAÚJO ASSIS

Heurísticas e metaheurísticas aplicadas ao Problema de Escalonamento Job-Shop

**Um algoritmo evolucionário híbrido baseado na
fertilização in vitro para solucionar problemas de
escalonamento job-shop**

Trabalho de Conclusão apresentado à Coordenação do Curso de Computação do Instituto de Informática da Universidade Federal de Goiás, como requisito parcial para obtenção do título de Bacharel em Computação.

Área de concentração: Otimização e Inteligência Artificial.

Orientador: Prof. Prof. Celso Gonçalves Camilo Júnior

Goiânia
2011

ÉWERTON CARLOS DE ARAÚJO ASSIS

Heurísticas e metaheurísticas aplicadas ao Problema de Escalonamento Job-Shop

**Um algoritmo evolucionário híbrido baseado na
fertilização in vitro para solucionar problemas de
escalonamento job-shop**

Trabalho de Conclusão apresentado à Coordenação do Curso de Computação do Instituto de Informática da Universidade Federal de Goiás como requisito parcial para obtenção do título de Bacharel em Computação, aprovada em 19 de Dezembro de 2011, pela Banca Examinadora constituída pelos professores:

Prof. Prof. Celso Gonçalves Camilo Júnior

Instituto de Informática – UFG

Presidente da Banca

Prof. Anderson da Silva Soares

Instituto de Informática – UFG

Resumo

Assis, Éwerton Carlos de Araújo. **Heurísticas e metaheurísticas aplicadas ao Problema de Escalonamento Job-Shop**. Goiânia, 2011. 72p. Relatório de Graduação. Instituto de Informática, Universidade Federal de Goiás.

Existem diversas soluções, heurísticas e metaheurísticas, para o problema de escalonamento job-shop. O presente trabalho tem por objetivo analisar algumas destas soluções encontradas na literatura, com foco primordial nas metaheurísticas e nos padrões encontrados nessas soluções. O objetivo não é realizar uma revisão de todas as soluções que foram desenvolvidas, mas as soluções recentes e que são preponderantemente referenciadas pela literatura por sua qualidade ou conceitos inovadores apresentados. A partir das análises feitas, propõem-se analisar os efeitos de uma nova solução metaheurística, um algoritmo evolucionário híbrido baseado na fertilização in vitro, também aplicado na mesma classe de problemas. A eficiência e efetividade do algoritmo desenvolvido é analisada sobre instâncias geralmente utilizadas pela literatura — instâncias disponibilizadas pela OR-Library. A solução evolucionária híbrida baseada na fertilização in vitro apresenta resultados de qualidade, comparáveis aos resultados obtidos por outras soluções encontradas na literatura. Alguns operadores de seleção e de variabilidade utilizados em soluções evolucionárias são analisados e obtém-se uma análise comparativa de configurações distintas, corroborando com o que é defendido pela literatura sobre a importância da escolha dos mecanismos de seleção e de variabilidade. A partir de 27 configurações distintas, 18 instâncias da OR-Library são exercitadas, obtendo-se um comparativo da solução evolucionária híbrida frente a uma solução evolucionária baseada em conceitos canônicos.

Palavras-chave

Otimização combinatória, problema de escalonamento job-shop, heurísticas, metaheurísticas, algoritmos genéticos, algoritmo evolucionário, computação evolucionária

Abstract

Assis, Éwerton Carlos de Araújo. **Heuristics and Metaheuristics applied to the Job-shop scheduling problem: a hybrid evolutionary algorithm based on in vitro fertilization to solve job-shop scheduling problems**. Goiânia, 2011. 72p. Relatório de Graduação. Instituto de Informática, Universidade Federal de Goiás.

There are several solutions, heuristics and metaheuristics, to solve the job-shop scheduling problem. The present work aims to analyse some of these solutions found in literature, with primary focus on metaheuristics and patterns found in these solutions. The goal is not to conduct a review of all the solutions that have been developed, but the recent ones that are referenced in the literature mainly for their quality and their innovative concepts presented. From the analysis made, we propose to analyze the effects of a new metaheuristic solution, a hybrid evolutionary algorithm based on in vitro fertilization, also applied in the same class of problems. The efficiency and the effectiveness of the developed algorithm is analyzed through instances commonly used in the literature — instances available from OR-Library. The evolutionary hybrid solution based on in vitro fertilization presents results of quality comparable to results obtained by other solutions in the literature. Some selection and variability operators used in evolutionary solutions are analyzed and a comparative analysis of different configurations is obtained; confirming what is advocated by the literature on the importance of the choice of selection mechanisms and variability. From 27 different configurations, 18 instances of the OR-Library are exercised, resulting in a comparative study between the hybrid evolutionary solution and a solution based on canonical evolutionary concepts.

Keywords

Combinatorial optimization, Job-shop scheduling problem, Heuristics, Metaheuristics, Genetic Algorithms, Evolutionary algorithm, Evolutionary computation

Sumário

Lista de Tabelas	7
Lista de Algoritmos	8
1 Introdução	9
1.1 Objetivo	9
1.2 Organização da monografia	10
2 Problemas de escalonamento job-shop	11
2.1 Definição do problema de escalonamento job-shop	11
2.2 Critérios de otimização	12
2.3 Critérios baseados na data de entrega	13
2.4 Critérios baseados em custos	13
2.5 Pressupostos em problemas de escalonamento job-shop	14
2.6 Extensões à definição tradicional	16
2.7 Classificação das agendas	16
3 Soluções heurísticas e metaheurísticas	18
3.1 Métodos e abordagens de otimização aplicados ao problema de escalonamento job-shop	19
3.1.1 Métodos heurísticos e metaheurísticos	19
3.1.2 Padrões identificados em soluções	20
4 Princípios de solução metaheurística evolucionária	22
4.1 Algoritmos genéticos	23
4.2 Estratégias evolutivas	23
4.3 Hibridização em algoritmos evolucionários	24
4.3.1 Algoritmo auxiliar paralelo baseado na fertilização in vitro	25
5 Algoritmo evolucionário híbrido baseado na fertilização in vitro	27
5.1 Representação dos indivíduos–soluções	31
5.1.1 População inicial	32
5.2 Operadores de variabilidade e seus efeitos sobre a solução evolucionária	32
5.2.1 Operadores de recombinação	33
5.2.2 Operadores de mutação	33
5.3 Operadores de seleção	34

6	Resultados obtidos	36
6.1	Efetividade da solução	65
6.2	Influência dos operadores de variabilidade e de seleção sobre a efetividade da solução	65
7	Conclusão	67
7.1	Trabalhos futuros	67
	Referências Bibliográficas	69

Lista de Tabelas

6.1	Configuração dos casos de experimentos	37
6.2	Resultados do caso de experimento 1	38
6.3	Resultados do caso de experimento 2	39
6.4	Resultados do caso de experimento 3	40
6.5	Resultados do caso de experimento 4	41
6.6	Resultados do caso de experimento 5	42
6.7	Resultados do caso de experimento 6	43
6.8	Resultados do caso de experimento 7	44
6.9	Resultados do caso de experimento 8	45
6.10	Resultados do caso de experimento 9	46
6.11	Resultados do caso de experimento 10	47
6.12	Resultados do caso de experimento 11	48
6.13	Resultados do caso de experimento 12	49
6.14	Resultados do caso de experimento 13	50
6.15	Resultados do caso de experimento 14	51
6.16	Resultados do caso de experimento 15	52
6.17	Resultados do caso de experimento 16	53
6.18	Resultados do caso de experimento 17	54
6.19	Resultados do caso de experimento 18	55
6.20	Resultados do caso de experimento 19	56
6.21	Resultados do caso de experimento 20	57
6.22	Resultados do caso de experimento 21	58
6.23	Resultados do caso de experimento 22	59
6.24	Resultados do caso de experimento 23	60
6.25	Resultados do caso de experimento 24	61
6.26	Resultados do caso de experimento 25	62
6.27	Resultados do caso de experimento 26	63
6.28	Resultados do caso de experimento 27	64

Lista de Algoritmos

5.1	Solução canônica (<i>GA/EV-JB</i>)	28
5.2	Solução híbrida (<i>IVF/EV-JB</i>)	29
5.3	Método de busca local ou mutação através de permutações	30
5.4	Roleta	35

Introdução

Soluções heurísticas e metaheurísticas são amplamente utilizadas como meio de solucionar problemas em otimização [14] [20]. Muitas delas são amplamente utilizadas, em suas formas canônicas ou híbridas, para solucionar instâncias de problemas de escalonamento job-shop [9]. Soluções baseadas em algoritmos evolucionários são comuns nesse panorama e são frequentemente comparadas a outras abordagens metaheurísticas com o fim de legitimar a qualidade das soluções propostas ou mesmo evidenciar limitações e superar deficiências [14].

Com a evidência de limitações e deficiências em algoritmos evolucionários, como o tempo de convergência e os diversos efeitos que a aleatoriedade causa sobre a solução algorítmica, é comum que novas abordagens, ditas híbridas, emergjam com o fim de superar essas deficiências e explorar qualidades específicas a cada abordagem algorítmica. Assim surge o algoritmo auxiliar paralelo baseado na fertilização in vitro, inicialmente aplicado em algoritmos genéticos, que tem por finalidade explorar os mecanismos de variabilidade dos algoritmos genéticos e superar as deficiências quanto a efetividade do algoritmo e o tempo de convergência do mesmo. Como essa abordagem híbrida é nova [8], considerações práticas sobre esta ainda são raras na literatura.

1.1 Objetivo

O presente trabalho tem por objetivo propor uma solução algorítmica baseada nos algoritmos evolucionários, notavelmente os algoritmos genéticos e as estratégias evolutivas, em conjunto com o algoritmo auxiliar paralelo baseado na fertilização in vitro, com o fim de analisar a aplicabilidade dessa abordagem híbrida em um contexto específico (problemas de escalonamento job-shop), e os efeitos que operadores de variabilidade e de seleção têm sobre a mesma.

1.2 Organização da monografia

O texto divide-se em capítulos, sendo o Capítulo 2 responsável por apresentar definições e conceitos relacionados a problemas de escalonamento job-shop; o Capítulo 3 apresenta as definições pertinentes a heurísticas e metaheurísticas e como estas abordagens estão sendo aplicadas para resolver problemas de escalonamento job-shop; o Capítulo 4 esboça os conceitos que norteiam a solução algorítmica proposta; o Capítulo 5 apresenta a solução algorítmica e as decisões de projeto envolvidas; o Capítulo 6 apresenta os resultados obtidos a partir da solução algorítmica e quais são as considerações observáveis a partir dos resultados; por fim, o Capítulo 7 apresenta a conclusão e trabalhos futuros.

Problemas de escalonamento job-shop

Problemas de escalonamento remontam desde a necessidade de organizar meios de produção para melhor satisfazerem a demanda de consumo dos mercados. A partir da Revolução Industrial houve uma crescente necessidade pela sistematização e otimização dos meios de produção, com fins de diminuir o tempo de produção e fazer melhor uso de recursos despendidos na manufatura. Problemas de escalonamento job-shop emergem nesse contexto de manufaturas, no qual tarefas ou produtos (*jobs*) devem ser processados por máquinas a fim de que todas as tarefas sejam concluídas. Contudo, problemas de escalonamento job-shop não se prendem a estes contextos de manufaturas; diversos problemas e contextos podem ser modelados e melhor compreendidos a partir dessa perspectiva.

2.1 Definição do problema de escalonamento job-shop

A definição formal de um problema de escalonamento job-shop tradicional [12] é feita a seguir: são dadas n tarefas (ou *jobs*) $\{J_1, J_2, \dots, J_n\}$ a serem processadas por m máquinas $\{M_1, M_2, \dots, M_m\}$ — cada tarefa deve ser processada por cada máquina uma única vez. O processamento de uma tarefa em uma máquina é denominado operação: a operação da tarefa i na máquina j é denotada por o_{ij} . Restrições tecnológicas determinam a ordem de processamento de cada tarefa através das m máquinas. As tarefas em um problema de escalonamento job-shop não compartilham uma mesma ordem de processamento através das máquinas, como acontece em problemas de escalonamento flow-shop, no qual todas as tarefas compartilham uma mesma ordem de processamento. O ambiente analisado em um problema de escalonamento job-shop é geralmente denotado por oficina (*workshop*).

Cada tarefa i (J_i) apresenta um tempo de processamento para cada máquina, i.e. para cada operação o_{ij} , operação da tarefa i na máquina j , há um tempo de processamento p_{ij} . Qualquer tempo de ajuste ou de carregamento (*set-up time*) da máquina para processar a operação o_{ij} é incluído em p_{ij} . Além do tempo de processamento para cada operação, cada tarefa i (J_i) pode apresentar um tempo de lançamento, denotado por r_i , que

determina a partir de quanto tempo ou em qual momento a tarefa J_i estará disponível para processamento pelas m máquinas da oficina.

A partir dessas restrições e definições surge a necessidade de construir um escalonador de tarefas de forma a agendar as n tarefas através das m máquinas de forma que o tempo total de processamento seja o menor possível. Além desse critério de otimização, outros critérios são usualmente estabelecidos, os quais serão apresentados a seguir. Com base nessa definição clássica do problema de escalonamento job-shop, outras definições também foram feitas, como extensão, as quais serão apresentadas em seguida.

O problema de escalonamento job-shop é considerado um problema numericamente intratável, NP-difícil (para $m \geq 2$) [12] e que apresenta um limite superior de $(n!)^m$ soluções possíveis; portanto, objetivamente, a enumeração de todas as soluções possíveis para instâncias com dimensão relevante (por exemplo, 10×15 ou 10 tarefas, 15 máquinas) é impraticável [33].

2.2 Critérios de otimização

A partir da definição preliminar do problema de escalonamento job-shop podemos extendê-la de forma a obter outros critérios de otimização. Essa definição básica geralmente é estendida da seguinte forma [12] [33]:

d_i é a data de vencimento ou a data de entrega da tarefa J_i , o tempo ideal para que a tarefa seja completada;

a_i é a janela de tempo no qual a tarefa J_i pode ser processada, denotada por $a_i = d_i - r_i$;

W_{ik} é o tempo de espera da tarefa J_i no processamento da operação k . Não necessariamente a k -ésima operação refere-se à máquina M_k ; refere-se à k -ésima operação da ordem de processamento da tarefa J_i (determinada pelas restrições tecnológicas da instância específica);

W_i é o tempo total de espera da tarefa J_i , ou $W_i = \sum_{k=1}^m W_{ik}$;

C_i é o tempo total de processamento da tarefa J_i , ou $C_i = r_i + \sum_{k=1}^m (W_{ik} + p_{ij(k)})$;

F_i é o tempo total de processamento da tarefa J_i sem levar em conta o tempo de lançamento desta, i.e. é o tempo total em que a tarefa passou pelas máquinas e, assim, permaneceu na oficina ($F_i = C_i - r_i$);

L_i é o atraso no processamento da tarefa J_i ($L_i = C_i - d_i$). Quando a tarefa é completada antes do previsto na data de entrega (antecipação), L_i será negativo. Portanto, a partir de L_i são definidas duas outras variáveis de análise: $T_i = \max\{C_i - d_i, 0\}$, o atraso da tarefa J_i ; e $E_i = \max\{d_i - C_i, 0\}$, a antecipação da tarefa J_i ;

I_j é o tempo de ociosidade da máquina M_j .

Estas definições têm como propósito formalizar os critérios de otimização usualmente estabelecidos para problemas de escalonamento job-shop. O critério básico, comumente adotado pela literatura em problemas de escalonamento [9], tem como objetivo minimizar o C_{max} (*makespan*), i.e. minimizar o tempo total da tarefa de maior tempo de processamento. Em problemas nos quais as tarefas não têm um tempo de lançamento, estas já estão todas preparadas para serem executadas pelas máquinas, $C_{max} = F_{max}$, sendo F_{max} o tempo total de processamento da tarefa que permaneceu por mais tempo na oficina (passou por todas as máquinas). Em contexto distinto, no qual algumas tarefas apresentam algum tempo de lançamento, podemos ter $C_{max} \neq F_{max}$; portanto, outro critério de otimização é a minimização do valor F_{max} , o tempo gasto pelas tarefas na oficina, referente à tarefa com maior tempo de processamento pelas máquinas. Estes dois critérios de otimização são denotados como critérios baseados no tempo de conclusão das tarefas [12]. Outros critérios são apresentados a seguir.

2.3 Critérios baseados na data de entrega

A partir da definição de L_i alguns critérios de otimização baseados na data de entrega podem ser estabelecidos [12], notadamente T_{max} e E_{max} , respectivamente, o atraso máximo e a antecipação máxima. A minimização do valor T_{max} é apropriada em contextos nos quais o atraso no processamento de tarefas apresenta alguma penalidade. Já no caso de minimização do valor E_{max} , esta abordagem é apropriada nos contextos em que há um benefício ao antecipar-se o processamento das tarefas.

Por fim, outra abordagem de minimização relativa à data de entrega é a minimização do número de tarefas atrasadas (n_T). Esta abordagem é apropriada em contextos críticos, nos quais tarefas atrasadas apresentam um alto valor de penalidade.

2.4 Critérios baseados em custos

Em alguns ambientes máquinas ociosas podem representar recursos ociosos ou, ainda, o processamento das tarefas está sujeito à disponibilidade de recursos para cada operação. Nestes contextos, outros critérios de otimização surgem. Esses critérios de otimização levam em conta as seguintes variáveis, para um determinado tempo t [12]:

$N_{w(t)}$ o número de tarefas esperando para serem processadas por alguma máquina ou tarefas que ainda não estão prontas para serem processadas;

$N_{p(t)}$ o número de tarefas sendo processadas no tempo t ;

$N_{c(t)}$ o número de tarefas completadas no tempo t ;

$N_{u(t)}$ o número de tarefas a serem ainda completadas no tempo t .

Assim, alguns critérios de otimização estabelecidos a partir destas variáveis são:

1. minimizar o número médio de tarefas esperando para serem processadas (\bar{N}_w) ou minimizar o número médio de tarefas não completadas (\bar{N}_u) de forma a minimizar os custos de inventário;
2. minimizar o número médio de tarefas completadas (\bar{N}_c), de forma a reduzir os custos dos bens produzidos;
3. maximizar o número médio de tarefas sendo processadas (\bar{N}_p) com o intuito de realizar um uso eficiente das máquinas. Essa maximização do valor \bar{N}_p relaciona-se à minimização do valor I_{max} , ou a minimização do tempo máximo que uma máquina fica ociosa.

2.5 Pressupostos em problemas de escalonamento job-shop

Pelas definições até então feitas é possível identificar a complexidade do problema de escalonamento job-shop e o quanto este pode se tornar complexo em ambientes reais, nos quais tarefas podem surgir aleatoriamente (ambientes dinâmicos) ou mesmo o tempo de processamento das tarefas pode não ser conhecido à priori (ambientes estocásticos) [12]. Assim, alguns pressupostos gerais são assumidos ao identificar e analisar problemas de escalonamento job-shop tradicionais [12]. Como será apresentado adiante, esses pressupostos podem ser ignorados ou modificados em extensões feitas ao problema de escalonamento job-shop tradicional.

cada tarefa é uma entidade as operações de uma determinada tarefa não podem ser processadas simultaneamente, i.e. cada operação constituinte de uma determinada tarefa deve ser processada isoladamente, sem paralelismo, em cada máquina;

não existe preempção enquanto uma operação é processada em uma máquina, esta continuará indisponível até que a operação seja concluída — não é admissível nesse contexto a interrupção do processamento;

cada tarefa é constituída de m operações distintas, uma para cada máquina esse pressuposto é uma consequência da definição anteriormente dada. Assim, temos que uma operação (ou, ainda, neste contexto, uma tarefa) não pode ser processada mais de uma vez em uma determinada máquina; e cada tarefa deve ser processada uma única vez em cada máquina;

tarefas não são canceladas cada tarefa deve ser processada pelas máquinas;

o tempo de processamento é independente da agenda construída o objetivo final do problema de escalonamento job-shop é a construção de uma agenda na qual as tarefas sejam escalonadas, i.e. uma ordem de processamento das tarefas será definida atendendo os critérios de otimização estabelecidos e as restrições envolvidas. Contudo, a agenda definida não influenciará os valores necessários para escalonar as tarefas; assim, o tempo de carregamento da tarefa ou o tempo de preparação para o processamento de uma tarefa não será influenciado pela agenda definida. Esse pressuposto caracteriza essa definição tradicional como independente da sequência ou agenda (*sequence-independent*);

tarefas podem ter que esperar para serem processadas este cenário pode ocorrer já que, como não é admissível a preempção ou o cancelamento de tarefas, e como o tempo de processamento (p_{ij}) é variável (entre tarefas e máquinas distintas), em alguns contextos tarefas podem ter que esperar para serem processadas em alguma máquina ocupada;

existe apenas um único tipo de cada máquina não existe diversas opções da mesma máquina na oficina analisada. Esse pressuposto elimina a possibilidade de evitar esperas entre as tarefas;

máquinas podem estar ociosas já que as diversas tarefas podem não ter concluído seu processamento em alguma das outras máquinas ou estão à espera de outras máquinas para serem processadas;

as máquinas não podem processar qualquer tarefa mais de uma vez

as máquinas da oficina estão sempre disponíveis para processamento das tarefas

as restrições tecnológicas são imutáveis e previamente conhecidas a ordem de processamento das operações de cada tarefa por cada máquina é previamente estabelecida e imutável ao longo do processo de escalonamento;

não existe aleatoriedade o número de tarefas, o número de máquinas, o tempo de processamento de cada operação, o tempo de lançamento de cada tarefa e outros valores usados para modelar um problema específico são previamente conhecidos e fixos.

Vale ressaltar que os pressupostos aqui estabelecidos aplicam-se todos ao problema de escalonamento job-shop tradicional, conforme definido em [12]. Sugestivamente, alguns destes pressupostos não são admissíveis em contextos e ambientes reais. As definições extendidas, apresentadas a seguir, geralmente modificam, estendem ou eliminam alguns destes pressupostos.

2.6 Extensões à definição tradicional

Com base na definição tradicional dos problemas de escalonamento job-shop, outras definições são feitas extendendo ou reinterpretando as restrições e definições aplicadas ao problema. Algumas dessas extensões são o problema de escalonamento job-shop flexível (FJSSP) e o problema de escalonamento job-shop multi-objetivo (MOJSSP).

Problema de escalonamento job-shop flexível (FJSSP) Essa extensão faz uma reinterpretação do que representa cada máquina e, conseqüentemente, aumenta o espaço de busca do problema. Além dos critérios de otimização analisados, um outro objetivo é acrescentado: além de determinar a ordem de execução das tarefas, o escalonador deve determinar em qual das máquinas determinada tarefa será executada em dado momento. Assim, não existe restrições tecnológicas quanto à ordem que as tarefas devem ser processadas pelas diversas máquinas. Por consequência, o espaço de busca é aumentado. [23] [35] [42] [46] [25] [45] [44]

Problema de escalonamento job-shop multi-objetivo (MOJSSP) Essa extensão na verdade é uma categorização de problemas de escalonamento job-shop que estabelecem mais de um critério de otimização no modelo do problema a ser solucionado. Portanto, existem algumas abordagens que podem ser estabelecidas para guiar a solução, como por exemplo estabelecer um sistema de pesos dentre os diversos critérios a fim de obter uma única função objetiva; ou a abordagem da eficiência à Pareto na resolução de problemas multi-objetivos. [1] [26] [46] [45] [31]

Estas extensões apresentadas não têm como propósito serem definitivas, já que outras definições podem ser obtidas a partir de outras considerações sobre os pressupostos e restrições apresentadas.

2.7 Classificação das agendas

Como foi supracitado, o espaço de busca do problema de escalonamento job-shop é consideravelmente grande até mesmo para pequenas instâncias do problema ($m \geq 2$). Assim, é encontrado na literatura uma categorização das soluções com base em alguns dos critérios de otimização analisados. A seguir são apresentadas as categorizações mais usuais:

agendas ativas são agendas nas quais não é possível escalonar tarefas (ou operações) para serem processadas antecipadamente sem violar algumas das restrições tecnológicas [6]. Outra definição apresenta agendas ativas como agendas nas quais não é possível ter ao menos uma tarefa terminando antecipadamente e nenhuma tarefa atrasada [30];

agendas semiativas são agendas nas quais existe tempo ocioso e que não pode ser eliminado, i.e. algumas máquinas estarão ociosas [33]. Outra definição afirma que agendas semiativas são aquelas nas quais tarefas (ou operações) não podem ser processadas antecipadamente sem modificar a ordem de processamento ou violar as restrições tecnológicas [6] [30];

agendas sem atraso são agendas nas quais não existe tempo ocioso, i.e. máquinas nunca estão ociosas [33];

agendas ótimas as agendas ótimas são parte de um subconjunto das agendas sem atraso e que minimizam o tempo de processamento de todas as tarefas, i.e. minimizam o tempo da tarefa com maior tempo de processamento (C_{max}) [33].

Essa categorização tem como objetivo clarificar quais são os objetivos a serem almejados em uma solução proposta, além de delimitar o espaço de busca que está sendo usado como ambiente de busca e quais as agendas que podem ser efetivamente construídas em determinada abordagem de solução.

Soluções heurísticas e metaheurísticas

O termo heurística origina-se do termo grego εὐρίσκω (*heuriskō*), que significa “a arte de descobrir novas estratégias (...) para solucionar problemas” (tradução livre) [38]. Dada a complexidade do problema de escalonamento job-shop, por exemplo, os métodos de solução trivialmente utilizados para outros problemas similares, ou o conhecimento utilizado para resolver outros problemas semelhantemente complexos (como a classe de problemas combinatoriais), não são facilmente empregados neste tipo de problema, dada a sua complexidade e características próprias e o fato de não ser solucionável em tempo polinomial (dado que o problema de escalonamento job-shop é NP-difícil). Portanto, heurísticas são desenvolvidas — a partir da análise das características do problema e de como contornar a necessidade de enumerar e verificar as soluções possíveis no espaço de busca — com o fim de apresentar soluções aceitáveis em tempo também aceitável. Geralmente são feitas análises das intâncias do problema, e os resultados obtidos através destas servem como estímulo para melhorar a solução heurística proposta. Soluções heurísticas são geralmente vistas como propícias a serem enganadas por ótimos locais [15].

Michel Gendreau, no livro *Handbook of Metaheuristics*, apresenta metaheurísticas como “métodos de solução que orquestram uma interação entre procedimentos de melhora local e estratégias de alto nível com o fim de criar um processo capaz de escapar de ótimos locais e realizar uma busca robusta em um espaço de busca” (tradução livre) [14]. Metaheurísticas “provêm resultados aceitáveis, em um tempo razoável, como solução de problemas difíceis e complexos, nas Ciências e na Engenharia” (tradução livre) [38]. O prefixo *meta* aplicado ao termo heurística denota que estas soluções são como modelos para que soluções heurísticas sejam facilmente construídas a partir de modificações mínimas nas metaheurísticas. O termo metaheurística foi inicialmente apresentado por Glover [15] [38].

Metaheurísticas auxiliam na construção de soluções para diversos problemas complexos por apresentarem um arcabouço geral e flexível (a partir de estratégias de alto nível), que tornam a customização da solução para necessidades específicas do problema relativamente simples (procedimentos de melhora local, como supracitado). Estas carac-

terísticas, analisadas a seguir, tornaram as metaheurísticas ferramentas importantes para solucionar problemas complexos e que apresentam instâncias de tamanho considerável.

3.1 Métodos e abordagens de otimização aplicados ao problema de escalonamento job-shop

Várias abordagens têm sido utilizadas para apresentar soluções ao problema de escalonamento job-shop [9]. Esses métodos de resolução são usualmente caracterizados como métodos exatos, métodos com regras de prioridade, métodos heurísticos e metaheurísticos e métodos da inteligência artificial [9]. Além do uso isolado desses métodos, é comum o uso híbrido destes com o propósito de melhorar uma abordagem específica [14]. As formas híbridas geralmente têm como forma a combinação de um método de busca global acompanhado por um método de busca local [9].

Realizar uma revisão sistemática e extensiva dos métodos de resolução atualmente empregados não é o fim deste trabalho; contudo, a partir da análise das heurísticas e metaheurísticas empregadas e citadas na literatura, os métodos de resolução analisados a seguir são aqueles citados como métodos heurísticos e metaheurísticos, foco primordial do presente trabalho.

3.1.1 Métodos heurísticos e metaheurísticos

Existem atualmente várias abordagens heurísticas e metaheurísticas com o intuito de apresentar soluções para o problema de escalonamento job-shop [9]. Algumas heurísticas e metaheurísticas notáveis são: algoritmos genéticos [16] [11] [18] [24] [25] [26] [19] [33] [41] [43] [44] [45], algoritmos meméticos [13], *particle swarm optimization* [46] [22], *simulated annealing* [22], colônia de formiga (*ant colony optimization*) [23] [33] [42], *variable neighborhood search* [1] [34] e busca tabu (*tabu search*) [35] [45]. Atualmente, é notável a tendência de uso híbrido dessas heurísticas e metaheurísticas [14] com o intuito de explorar as vantagens e minimizar as desvantagens de cada método [9].

O uso de métodos heurísticos e metaheurísticos, também denominados métodos aproximados [33], em contrapartida aos métodos exatos (ou métodos de otimização clássicos, como *branch-and-bound* [12] [2], *cutting-plane* e programação inteira [12]) tem como motivador a qualidade das soluções obtidas, embora não garantidamente ótimas. Portanto, são algoritmos relativamente efetivos, além de serem soluções eficientes, quando comparado aos métodos exatos [9]. Os métodos exatos são ineficientes, embora efetivos, na resolução de instâncias de problemas com dimensão considerável (por exemplo, 15×15 ou 15 tarefas e 15 máquinas) [9].

Os métodos aproximados são geralmente classificados como técnicas construtivas ou de busca local [33]. Aqueles têm o propósito de obter uma solução final a partir de um espaço de solução vazio e, iterativamente, melhorar essa solução inicial (ou soluções iniciais, no contexto das abordagens populacionais). A busca local tem como propósito melhorar uma solução completa inicial na tentativa de obter uma solução ainda melhor. A performance e qualidade dos métodos aproximados é geralmente analisada a partir dos resultados obtidos experimentalmente [20] [33].

3.1.2 Padrões identificados em soluções

Alguns padrões na construção de soluções heurísticas e metaheurísticas foram identificados a partir das análises feitas com os artigos selecionados:

1. Ainda que existam tentativas de propor novas formas de representação das soluções que os métodos metaheurísticos trabalharão, como uma codificação cromossômica tridimensional ou matricial para algoritmos genéticos [19] [43] [44], obtém-se resultados de qualidade evidente, eficiente e relativamente efetiva com uma representação cromossômica a partir de chaves aleatórias, ou ostensivamente conhecidas como *random-keys* [1] [16] [34] [22]. A principal vantagem da representação por chaves aleatórias é que todas as soluções-indivíduos, do método metaheurístico populacional (algoritmo genético, por exemplo), são indivíduos válidos em todas as aplicações dos operadores de recombinação e mutação; nenhum esforço adicional é necessário para tornar uma solução-indivíduo válida [16]. Conquanto exista o esforço computacional extra para decodificar a representação da solução para uma agenda (no caso do problema de escalonamento job-shop), com determinada ordem de processamento das operações das tarefas. Essa representação, ou codificação da solução, tem sido utilizada sob o título de “representação baseada em operação” (*operation-based representation; a set of ordered operations; operation order representation*) justamente por centrar-se na ordem de processamento das operações para cada tarefa da oficina [34] [41] [46] [35] [33].
2. O uso de operações de permutação como operador de busca local ou como operador de mutação em algoritmos genéticos, ou demais soluções metaheurísticas, é relativamente comum nas soluções analisadas [22] [45] [41] [31] [26] [46] [1] [34]. Os operadores de permutação são utilizados por tratar-se de um problema combinatorial e seu uso é suscitado pela representação usualmente escolhida. Contudo, é relativamente comum que algumas soluções optem por algoritmos de construção de agendas (*constructive algorithms*) como operador de busca local [16]; ou mesmo como um método isolado de resolução, embora relativamente ineficaz [12]. Os operadores de permutação encontrados são: *a*) inserção; *b*) inversão (geralmente de ele-

mentos adjacentes); *c*) permutação ou troca de elementos em posições distintas; e *d*) movimento de longa distância (mover uma sequência de elementos a uma determinada posição ou direção). Esses operadores são encontrados com nomes distintos em alguns artigos, mas pela definição de suas operações é possível determinar qual operador está sendo utilizado.

3. Métodos metaheurísticos em suas versões canônicas são raramente utilizados como solução de qualidade para problemas de escalonamento job-shop — com base nas soluções analisadas através dos artigos selecionados. As versões canônicas de métodos metaheurísticos são geralmente utilizadas como método comparativo para os métodos híbridos, estabelecendo-se uma base de comparação para as soluções que são propostas como extensões, ou híbridas.
4. Conquanto algumas soluções sejam avaliadas a partir de instâncias do problema de escalonamento job-shop construídas especificamente para a solução proposta, ou obtidas de oficinas reais [42] [23] [46] [35] [34] [44] [43] [26] [25], as 82 instâncias disponíveis na OR-Library [5] e as 80 instâncias de Taillard [37] são frequentemente utilizadas como meios para estabelecer-se a qualidade da solução proposta frente aos resultados obtidos por outras soluções [32] [29] [21] [11] [18] [41] [45] [22] [16]. O uso de instâncias reconhecidas pela literatura garante que novas soluções propostas sejam comparadas com outras soluções já estabelecidas.

Princípios de solução metaheurística evolucionária

Problemas em otimização são uma classe de problemas computacionais não facilmente tratáveis, de forma que várias soluções baseadas em metaheurísticas foram criadas com o fim de melhor solucioná-las [14], quando aplicável. Uma das metaheurísticas mais utilizadas como meio para solucionar problemas em otimização são os algoritmos genéticos; seja em sua forma canônica, desenvolvida por Holland [17] a partir de trabalhos iniciais de outros autores [10], ou em soluções híbridas e conceitualmente modificadas [14]. A solução algorítmica apresentada neste trabalho tem por base os algoritmos genéticos, além de conceitos encontrados em outros algoritmos evolucionários. Além disso, a solução proposta é conflitada com a versão canônica dos algoritmos genéticos encontrada na literatura, com alguns conceitos modificados [20] [10], como a representação e aplicabilidade dos operadores de variabilidade.

Ao lado da programação evolucionária e das estratégias evolutivas, os algoritmos genéticos fazem parte um conceito ainda maior: algoritmos evolucionários [20]. Embora tenham uma origem temporal em comum, inicialmente os algoritmos evolucionários tentavam solucionar classes de problemas específicos; portanto, algumas divergências conceituais, mesmo que mínimas, eram comuns quando os algoritmos evolucionários eram aplicados [20]. Com o advento do termo algoritmos evolucionários e do arcabouço propiciado pela inteligência computacional, as formas híbridas dos algoritmos evolucionários passaram a coexistir às versões canônicas daqueles algoritmos [20]. Portanto, embora a solução proposta neste trabalho seja baseada em algoritmos genéticos, conceitos dos demais algoritmos evolucionários, em especial das estratégias evolutivas, são levados em consideração ao propor uma nova solução algorítmica.

As soluções baseadas em algoritmos evolucionários se distinguem das demais soluções metaheurísticas por serem algoritmos baseados na representação da solução do problema como um indivíduo (solução-indivíduo) e por influenciar-se de conceitos biológicos como evolução, aptidão, recombinação e mutação genética e desenvolvimento geracional de uma população.

4.1 Algoritmos genéticos

A partir de uma população de indivíduos–soluções aleatoriamente criada, na qual cada indivíduo é representado por uma cadeia de *bits*, os algoritmos genéticos em sua versão canônica desenvolvem a seguinte trajetória evolutiva sobre uma população:

1. são selecionados indivíduos–soluções da população corrente — através de um mecanismo de seleção de indivíduos que leve em consideração a aptidão (*fitness*) dos indivíduos daquela população — para gerar novos descendentes;
2. novos indivíduos são gerados a partir de um mecanismo de recombinação genética (*crossover*);
3. o novo indivíduo gerado é mutado a partir de um mecanismo de mutação;
4. a população corrente é substituída pelos descendentes gerados, dando origem a uma nova geração.

Essa trajetória evolutiva geralmente é mantida até que um número limite de gerações seja alcançadas, embora outras abordagens podem ser determinadas, como um limite temporal, ou mesmo como aquela população evolui temporalmente — se há ganho ou não a cada ciclo geracional. Um meio de mensuração da aptidão dos indivíduos deve ser provido ao algoritmo, de forma que o algoritmo consiga estabelecer alguns parâmetros de análise e comparação dos indivíduos a cada geração. Esse meio de mensuração da aptidão é geralmente obtido a partir da definição do problema a ser tratado.

Assim, os algoritmos genéticos, como uma solução metaheurística, tem como finalidade ser uma solução para problemas de otimização de fácil implementação e utilização: a partir de uma representação binária das soluções e de um meio de mensuração da aptidão dos indivíduos — uma forma de se estabelecer a qualidade daquela solução–indivíduo em questão —, o algoritmo genético constrói um conjunto de soluções a fim de obter novos indivíduos (soluções), percorrendo um determinado espaço de busca, a partir da representação dos indivíduos daquela população.

4.2 Estratégias evolutivas

Diferentemente dos algoritmos genéticos que surgiram como uma solução metaheurística para diversos problemas de otimização — bastava prover ao algoritmo uma forma de mapear a representação do indivíduo de uma solução para um problema específico para uma representação em cadeia binária; e uma função de mensuração da qualidade/aptidão daquele indivíduo (*fitness*) —, as estratégias evolutivas surgiram como uma metaheurística específica para solucionar problemas de otimização de variáveis contínuas

[20]. Indivíduos são representados através de vetores de valores reais, nos quais cada dimensão armazena o valor de uma das variáveis a serem otimizadas. Algoritmos genéticos, em seus conceitos e considerações canônicos, têm suas deficiências ao tratar de otimização de números reais, já que, na representação dos indivíduos, sempre limita-se a capacidade representativa do espaço de busca.

Outras diferenças conceituais são notáveis [20]: 1) o mecanismo ou operador de variabilidade das estratégias evolutivas baseava-se na reprodução assexuada; a cada geração novos indivíduos eram gerados a partir de mutações, obrigatoriamente. Diferentemente dos algoritmos genéticos, que tinham um mecanismo de reprodução sexuada (recombinação) e um mecanismo secundário de mutação (conceitualmente assexuado). Ambos os mecanismos de variabilidade dos algoritmos genéticos são tratados como aplicáveis ou não na geração de um novo indivíduo; 2) há um foco importante no controle do *step-size*, ou à maneira que o algoritmo percorre o espaço de busca, se será dado passos largos ou curtos nesse espaço de busca — controle feito a partir do operador de variabilidade (mutação); 3) de início as estratégias evolutivas não eram uma metaheurística populacional, mas centrada num único indivíduo, que dava origem a vários indivíduos, mas apenas um, o melhor dentre os indivíduos gerados e o pai, era selecionado para continuar existindo na próxima geração. Posteriormente, verificou-se a importância do conceito populacional (“busca paralela adaptativa” [20]).

A principal influência das estratégias evolutivas na concepção da solução algorítmica apresentada neste trabalho trata-se na importância do mecanismo de mutação dos indivíduos gerados; na importância que existe em sempre perturbar aquelas soluções de forma a gerar novas possibilidades de análise no espaço de busca; e na importância que é dado ao melhor indivíduo que já existiu dentre as populações, conceito empregado nos algoritmos genéticos sob o título de “elitismo”. Considerações a respeito destes conceitos são dados com maiores detalhes quando a solução algorítmica for apresentada, no Capítulo 5.

4.3 Hibridização em algoritmos evolucionários

A origem dos algoritmos evolucionários remonta à década de 1960, quando conceitos evolutivos foram implementados em algoritmos como forma de solucionar problemas de otimização [20]. Cada algoritmo evolucionário surge em um contexto específico de utilização e com características próprias que os tornaram boas *metasoluções* para determinadas classes de problemas em otimização [20] [14]. Assim, nas décadas de 1980 e 1990 surge uma expansão do uso destes algoritmos para outras classes de problemas, o que nem sempre resultava em soluções de qualidade. Outrossim, novas

abordagens metaheurísticas também surgiram, o que contribuiu para a origem das formas híbridas de solução metaheurísticas [20] [14].

As soluções híbridas surgem como uma forma de melhor aproveitar a eficiência e efetividade das metaheurísticas, em foco os algoritmos evolucionários, como uma forma de melhor balancear as características de exploração do espaço de busca (*exploration*) e realizar melhoras discretas nos indivíduos–soluções desenvolvidos nas soluções evolucionárias (*exploitation*) [7] [14]. Assim, operadores ou mecanismos de variabilidade são criados e conceitos adaptados para melhor aproveitar as características dos algoritmos evolucionários como solução. Exemplos de modificações são o IGA (*Immune Genetic Algorithm*) [25], técnicas de nicho [7], auto-adaptação [20], sistemas coevolutivos [20], dentre outros. Como apresentado, grande parte das soluções metaheurísticas analisadas que são aplicadas em problemas de escalonamento job-shop são abordagens híbridas.

Uma abordagem híbrida nova é o algoritmo auxiliar paralelo (AAP) baseado na fertilização in vitro, inicialmente aplicado em algoritmos genéticos, a qual será apresentada a seguir. O AAP tem por principal objetivo aproveitar melhor as informações armazenadas na representação genética dos indivíduos–soluções dos algoritmos evolucionários, notadamente em algoritmos genéticos. Esse melhor aproveitamento tem ainda por finalidade realizar um melhor controle de convergência do algoritmo sem denegrir a qualidade final das soluções obtidas.

4.3.1 Algoritmo auxiliar paralelo baseado na fertilização in vitro

O algoritmo auxiliar paralelo (AAP) baseado na fertilização in vitro tem por base uma consideração simples: os melhores indivíduos–soluções de uma população (melhor em termos de sua aptidão frente aos demais indivíduos) tem em sua estrutura genética/cromossômica características que os tornam indivíduos de notável qualidade. Assim, com o fim de melhor aproveitar estas características, é feita uma seleção desses melhores indivíduos e uma recombinação dentre eles a fim de gerar um melhor indivíduo. Com forte influência dos conceitos biológicos da fertilização in vitro, o AAP seleciona os genes dos pais a fim de construir um novo indivíduo de qualidade comparável a de seus pais [7].

O AAP tem um fluxograma conceitualmente simples, subdivide-se nas fases de coleta, manipulação genética e transferência. A **coleta** tem por finalidade selecionar os indivíduos de maior qualidade da população corrente de forma a prepará-los para a fase seguinte, de **manipulação genética**. Nessa fase os melhores indivíduos selecionados têm suas características genéticas manipuladas de forma a compor um novo indivíduo. O novo indivíduo gerado é avaliado frente a todos os demais indivíduos da população, incluindo seus pais; caso seja um indivíduo de maior qualidade, este passa a fazer parte da população

através de uma estratégia de **transferência** [7].

Maiores considerações a respeito do AAP serão dadas na apresentação da solução evolucionária desenvolvida neste trabalho, Capítulo 5.

Algoritmo evolucionário híbrido baseado na fertilização in vitro

A partir das características identificadas da classe de problemas de escalonamento job-shop e com o intuito de obter uma comparação de uma abordagem híbrida frente a uma abordagem canônica, foram desenvolvidas duas propostas de solução baseadas em algoritmos genéticos: *a)* uma solução centrada nos princípios e pressupostos desenvolvidos em torno dos algoritmos genéticos, que têm se mostrado aceitos pela comunidade especializada, notavelmente aqueles relacionados à computação evolucionária [20]; e *b)* uma solução híbrida, através de um algoritmo auxiliar paralelo inspirado na fertilização in vitro [8], denominado algoritmo evolucionário híbrido baseado na fertilização in vitro como solução de problemas de escalonamento job-shop (IVF/EV-JB). Ambas as soluções foram comparadas com configurações distintas em termos de operadores de seleção, operadores de recombinação e a aplicação arbitrária de um método de busca local, obtendo-se assim 27 casos de experimentos.

A solução canônica foi construída conforme as descrições feita por De Jong em *Evolutionary computation: a unified approach* [20], contudo sem perder as referências iniciais de Holland em *Adaptation in Natural and Artificial Systems* [17]. De Jong defende uma abordagem mais flexível dos algoritmos evolucionários, ainda que seja levado em consideração os paradigmas canônicos destes algoritmos e a importância de cada aspecto defendido pelos algoritmos evolucionários. A solução proposta por neste trabalho, portanto, a partir dos algoritmos genéticos, está em consonância com o que é esperado desta classe de algoritmos: a partir de um conjunto de indivíduos—soluções aleatoriamente construídos, é feita uma recombinação destes indivíduos, como um importante mecanismo de variabilidade, e mutações são realizadas com o propósito de obter-se uma exploração agressiva do espaço de busca, conforme delineado no Algoritmo 5.1.

Algoritmo 5.1: Solução canônica (GA/EV-JB)

Entrada: População de tamanho μ aleatoriamente construída

```

1 contador_gerações  $\leftarrow$  0
2 repeat
3   Seleciona  $\rho = \mu/2$  pais para reprodução
4   for  $i \rightarrow 0$  to  $\rho$  do
5     descendente  $\leftarrow$  Recombinação do pai  $2 * i$  com o pai  $2 * i + 1$ 
6     Mutação realizada sobre descendente (busca local)
7     Adição de descendente a um conjunto  $\lambda$  de descendentes
8   end
9   O conjunto  $\lambda$  de descendentes gerados substitui a geração parental e
   passa a ser a população corrente
10  Elitismo: o pior indivíduo dentre os descendentes gerados é substituído
   pelo melhor indivíduo da geração parental
11  contador_gerações  $\leftarrow$  contador_gerações + 1
12 until contador_gerações < máximo_gerações

```

Os algoritmos genéticos, conforme apresentado anteriormente, assumem um mecanismo de controle dos operadores de recombinação e de mutação através da probabilidade de acontecer ou não estes mecanismos; ou seja, há um controle na aplicação ou não destes operadores sobre os indivíduos-soluções da população. A solução inicialmente proposta foi construída com estes princípios; contudo, durante a fase de afinação da solução, verificou-se a perda da qualidade das soluções obtidas quando os operadores de variabilidade eram estabelecidos sobre a população a partir de probabilidades. Assim, para efeito do presente trabalho, a probabilidade de ocorrer a recombinação de dois indivíduos e a mutação de um novo indivíduo gerado é de 100%. Além destes mecanismos de variabilidade, o conceito de elitismo está presente nas duas soluções consideradas: o pior indivíduo dentre os descendentes gerados é substituído pelo melhor indivíduo da geração parental. Desta forma garante-se que não haverá a perda de indivíduos-soluções de qualidade a cada geração.

Algoritmo 5.2: Solução híbrida (IVF/EV-JB)

Entrada: População de tamanho μ aleatoriamente construída

```

1 contador_gerações  $\leftarrow$  0
2 repeat
3   superpais  $\leftarrow$  Os melhores  $\rho' = \mu * 25\%$  indivíduos da geração corrente
4   embrião  $\leftarrow$  Manipulação genética dos superpais
5   Seleciona  $\rho = \mu/2$  pais para reprodução
6   for  $i \rightarrow 0$  to  $\rho$  do
7     descendente  $\leftarrow$  Recombinação do pai  $2 * i$  com o pai  $2 * i + 1$ 
8     Mutação realizada sobre descendente (busca local)
9     Adição de descendente a um conjunto  $\lambda$  de descendentes
10  end
11  O conjunto  $\lambda$  de descendentes gerados substitui a geração parental e
    passa a ser a população corrente
12  Elitismo: o pior indivíduo dentre os descendentes gerados é substituído
    pelo melhor indivíduo da geração parental
13  Transferência do embrião para a população de novos indivíduos
14  contador_gerações  $\leftarrow$  contador_gerações + 1
15 until contador_gerações < máximo_gerações

```

A abordagem híbrida, conforme delineado no Algoritmo 5.2, obtida a partir de um algoritmo auxiliar paralelo (AAP) baseado na fertilização in vitro [8], tem por base uma extensão que tem por propósito propiciar à solução evolucionária mecanismos que garantam a qualidade genética dos indivíduos, através da promoção destas características, e ter um maior controle da convergência do algoritmo, tornando-a mais rápida e de qualidade, sem denegrir os resultados que possam ser obtidos por essa solução [8] [7].

O AAP ocorre, conceitualmente, de forma paralela ao fluxo principal dos algoritmos genéticos, na fase de estabelecimento de uma nova geração populacional. O AAP dividi-se em três fases distintas: 1) **coleta** — esta fase tem por propósito selecionar os melhores indivíduos da população, já que esta abordagem algorítmica assume que os melhores indivíduos da população apresentam, geneticamente, as melhores características daquele conjunto de indivíduos-soluções. Nos experimentos desenvolvidos, os 25% melhores indivíduos da população são coletados nessa fase; 2) **manipulação genética** — a partir dos indivíduos previamente coletados é feita uma manipulação genética sobre estes de forma a gerar novos indivíduos através da avaliação, alteração e recombinação daqueles. Os novos indivíduos são gerados a partir de um superpai e os demais indivíduos da coleta; 3) **transferência** — caso o processo de manipulação genética gere melhores indivíduos do que o melhor indivíduo corrente, esses são transferidos para a nova geração

populacional [8]. A estratégia de transferência utilizada é a substituição do pior indivíduo da geração parental, conforme delineado no Algoritmo 5.2.

Os operadores de seleção utilizados são o ranqueamento linear [20], o torneio [10] [27] e a seleção proporcional à aptidão dos indivíduos da população (*fitness proportional selection*) [20]. Os operadores de recombinação individualmente aplicados e analisados foram a recombinação de 1– e n –pontos [20] e a recombinação uniforme (*parameterized uniform crossover*) [20] [16].

Os operadores de mutação utilizados são dois operadores que são usualmente utilizados em problemas combinatorias e que já foram testados por outros textos relacionados a problemas de escalonamento job-shop. Os operadores de mutação são inicialmente testados separadamente em 18 casos de experimentos (1–18) e posteriormente usados em conjunto nos 9 casos de experimentos finais (19–27). O operador de permutação [20] [22] realiza modificações na representação dos indivíduos, de forma a perturbar as representações destes indivíduos obtendo-se variações que contribuam para uma melhor prospecção do espaço de busca, em conformidade com o que é reconhecido pela literatura como um operador de mutação [20].

Algoritmo 5.3: Método de busca local ou mutação através de permutações

Entrada: indivíduo a ser mutado

```

1  número_aleatório  $\leftarrow$  distribuição_normal(0, 100)
2  if número_aleatório  $\leq$  ProbExchange then
3    | Operador de troca (indivíduo)
4  else
5    | if ProbExchange < número_aleatório  $\leq$  ProbExchange + ProbInsertion then
6    | | Operador de inserção (indivíduo)
7    | else
8    | | if ProbExchange + ProbInsertion < número_aleatório  $\leq$ 
9    | | | ProbExchange + ProbInsertion + ProbInversion then
10   | | | Operador de inversão (indivíduo)
11   | | | else
12   | | | Operador de deslocamento (indivíduo)
13   | | end
14  end

```

O operador de permutação, conforme descrito no Algoritmo 5.3, realiza no máximo um tipo de modificação no indivíduo: *a) a operação de troca (exchange)* — a partir de dois valores, i e j , em um vetor com $n \times m$ posições ($0 < i < j \leq n \times m$), os

valores nas posições i e j são trocados entre si; *b*) a **operação de inserção** (*insertion*) — a partir de dois valores, i e j , em um vetor com $n \times m$ posições ($0 < i < j \leq n \times m$), o valor na posição i é inserido na posição j , deslocando os valores nas posições menores ou iguais a j para a esquerda, de forma a possibilitar a inserção do valor em i sem perder nenhum valor do vetor; *c*) a **operação de inversão** (*inversion*) — a partir de dois valores, i e j , em um vetor S com $n \times m$ posições ($0 < i < j \leq n \times m$), o segmento de valores entre as posições i e j são invertidos de forma que o vetor inicial $S = \langle \dots, s_i, s_{i+1}, \dots, s_{j-1}, s_j, \dots \rangle$ é transformado no vetor $S' = \langle \dots, s_j, s_{j-1}, \dots, s_{i+1}, s_i, \dots \rangle$; *d*) a **operação de deslocamento** (*shifting*) — a partir de três valores, i , j e k , em um vetor com $n \times m$ posições ($0 < i < j \leq n \times m$; $k < i$ ou $k > j$ e $0 < k < n \times m$), o segmento de valores entre as posições i e j são deslocados à posição k do vetor. Esse operador de permutação foi construído com base em artigos encontrados na literatura e com a experiência benéfica obtida por solução específica baseada em *particle swarm optimization* [22]. Os valores das probabilidades determinados para análise da efetividade da solução proposta são: $Prob_{Exchange} = 35\%$, $Prob_{Insertion} = 35\%$, $Prob_{Inversion} = 10\%$ e $Prob_{Shifting} = 20\%$. Estes valores foram determinados experimentalmente, com base nas escolhas feitas em artigo específico [22].

O segundo operador de mutação utilizado é o operador de geração aleatória [16]. O propósito desse operador é substituir 20% dos piores indivíduos da população por novos indivíduos construídos aleatoriamente a partir de uma distribuição normal. Assim, obtemos novos valores aos alelos dos indivíduos, já que, conforme descrito a seguir, os indivíduos são representados como cadeias de números reais.

5.1 Representação dos indivíduos–soluções

Os indivíduos–soluções são representados utilizando a representação *random keys* [4] [3] [36], geralmente utilizada para esse tipo de problema por ser uma representação simples, quando comparada a outras representações, como a matricial/tridimensional [44] [43], e efetiva, por deixar a solução algorítmica livre de algumas preocupações à mais, como a construção de operadores de recombinação e mutação que sempre gerem indivíduos válidos.

A representação do indivíduo através das *random keys* é feita a partir de um vetor de $n \times m$ posições — n denota o número de tarefas e m o número de máquinas. Cada dimensão do vetor é iniciada com valores reais a partir de uma distribuição normal $N(0, n \times m)$. Essa representação é centrada nas operações: o vetor como solução descreve a ordem a qual as operações na oficina serão processadas. A fim de descobrir cada uma das operações a partir da representação inicial, em valores reais, são feitas as seguintes operações — a título de exemplificação, tomemos como exemplo o vetor de 8 posições, com $n = 2$ e $m = 4$, $\langle 0.2, 0.5, 1.8, 6.7, 3.3, 2.4, 3.5, 2.4 \rangle$: 1) a partir do vetor

inicialmente construído através da distribuição normal, é feito um ranqueamento dos valores de cada dimensão, de forma a ordenar as dimensões do vetor em relação às demais dimensões. Então, no vetor de 8 posições tomado como exemplo, o ranqueamento de cada dimensão resultaria em um vetor com a seguinte configuração $\langle 1, 2, 3, 8, 6, 4, 7, 5 \rangle$; 2) após o ranqueamento das dimensões do vetor é necessário identificar cada uma das tarefas que o vetor-solução está referenciando e estabelecer a ordem de processamento de cada uma das operações. Para a dimensão i do vetor S ranqueado, temos $S'_i = (S_i \bmod n) + 1$, S' o vetor final com a sequência de processamento das operações. No vetor tomado como exemplo teríamos, para $n = 2$, a seguinte configuração final: $\langle 2, 1, 2, 1, 1, 1, 2, 2 \rangle$; 3) por fim, a partir da sequência de processamento das operações, obtemos o vetor $\langle o_{2,1}, o_{1,1}, o_{2,2}, o_{1,2}, o_{1,3}, o_{1,4}, o_{2,3}, o_{2,4} \rangle$, com $o_{i,k}$ a operação da tarefa i na k -ésima máquina de sua configuração tecnológica — se a tarefa 2 fosse inicialmente processada na máquina 4, $o_{2,1}$ refere-se à operação da tarefa 2 na máquina 4.

5.1.1 População inicial

Como é de costume em soluções baseadas em algoritmos evolucionários [20] [16], os indivíduos-soluções da população inicial são criados a partir de distribuições normais. Cada dimensão do vetor-indivíduo é iniciada com o valor da distribuição $N(0, n \times m)$, sendo n o número de tarefas e m o número de máquinas da oficina. O tamanho da população (μ) também é influenciado pelas configurações da oficina (problema) em questão: $\mu = 2 \times n \times m$.

5.2 Operadores de variabilidade e seus efeitos sobre a solução evolucionária

Considerações a respeito da influência dos operadores de variabilidade em algoritmos evolucionários são reconhecidos e estudados na literatura [20] [27]. No presente trabalho, além da análise do comportamento do algoritmo híbrido frente a abordagem canônica, foi feita uma análise dos efeitos que os operadores de variabilidade utilizados causaram sobre a efetividade da solução proposta.

Os operadores de variabilidade dividem-se em operadores de recombinação, os quais, a partir de um ou mais indivíduos, geram um novo indivíduo; e os operadores de mutação, que a partir de um indivíduo gerado por recombinação obtém-se um indivíduo com características novas naquela população, como será explicado a seguir.

5.2.1 Operadores de recombinação

Os operadores de recombinação utilizados são aqueles usualmente encontrados em soluções baseadas nos algoritmos genéticos. Conquanto a representação dos indivíduos-soluções seja feita a partir de um vetor de números reais (*random-keys*), conceitualmente a recombinação permanece idêntica a uma solução baseada em indivíduos representados através de cadeias binárias.

recombinação de 1-ponto a partir de dois indivíduos é gerado um novo indivíduo. Com esse fim, é feita a recombinação dos genes dos dois indivíduos, doravante denominados pais. Inicialmente é feita a escolha aleatória de um ponto de corte através de uma distribuição discreta a fim de escolher qual ponto de corte determinará a composição do novo indivíduo. Seja p e m os vetores-pais e f o vetor-filho, o novo indivíduo a ser gerado, e i o ponto de corte ($|p| = |m| = |f| = n, 0 \leq i < n$), o vetor-filho f será composto da seguinte forma: $f = \langle p_0, \dots, p_{i-1}, m_i, m_{i+1}, \dots, m_n \rangle$;

recombinação de n -pontos de forma semelhante à recombinação de 1-ponto, a recombinação de n -pontos determina n pontos distintos como pontos de corte para compor o novo indivíduo. Após análises dos efeitos dos valores de n , a solução final tem um valor de $n = 4$, portanto fixo. O termo correto, assim, seria recombinação de 4-pontos;

recombinação uniforme a recombinação uniforme funciona de forma distinta; não são estabelecidos pontos de corte. Ainda assim, a partir de dois indivíduos é gerado um novo indivíduo. Para compor esse novo indivíduo, cada gene do vetor-indivíduo deve ser escolhido arbitrariamente entre os vetores-pais; essa escolha é feita a partir do lançamento de uma moeda “viciada”: é feita uma distribuição normal, $m \leftarrow N(0, 100)$. Caso $m < 70$, é cara, o contrário é coroa. Assim, para cada gene, determina-se cara-ou-coroa, gene no pai ou gene da mãe.

5.2.2 Operadores de mutação

Como a representação dos indivíduos-soluções é feita a partir de vetores de números reais, os operadores de mutação usualmente utilizados em algoritmos genéticos não são utilizados na solução algorítmica construída. Os operadores de mutação utilizados são baseados em experiências de qualidade encontradas na literatura [16] [22].

permutação conforme delineado anteriormente, esse operador de mutação tem por finalidade reorganizar as dimensões do vetor de forma aleatória, perturbando os novos indivíduos-soluções gerados. Essa perturbação em nível genotípico é revertido em nível fenotípico para uma perturbação na ordem de processamento das operações das tarefas;

geração aleatória esse operador tem por influência as estratégias evolutivas [16] e tem por finalidade prover à população de indivíduos–soluções novos valores genotípicos, o que acarretará em perturbações em nível fenotípico. Os 20% piores indivíduos da população são substituídos por novos indivíduos aleatoriamente criados, de forma análoga à população inicial (Seção 5.1.1).

5.3 Operadores de seleção

Os operadores de variabilidade têm sua influência na forma como novos indivíduos serão gerados e como o espaço de busca do problema de otimização será percorrido, a partir das escolhas feitas pela representação dos indivíduos–soluções e como os operadores de variabilidade são capazes de evoluir/transformar essas representações. Por outro lado, os operadores de seleção determinam as escolhas a longo prazo que o algoritmo fará de quais características–indivíduos permanecerão naquela população a cada geração. Portanto, a interação entre os operadores de variabilidade e os operadores de seleção são fatores importantes na análise da efetividade de uma solução baseada em algoritmos evolucionários.

Para efeito de análises foram utilizados três operadores de seleção distintos [20] [10]:

seleção por ranqueamento linear operador de seleção mais simples e com maior pressão seletiva, a seleção por ranqueamento linear organiza os indivíduos–soluções dos melhores aos piores; os melhores indivíduos são sempre escolhidos em detrimento dos piores indivíduos da população;

seleção proporcional à aptidão a seleção proporcional à aptidão (*fitness*) tem pressão seletiva média e garante a todos os indivíduos a oportunidade de serem selecionados. Inicialmente é feito o somatório da aptidão dos indivíduos da população: seja n o tamanho da população, f_i a aptidão do indivíduo i , $1 \leq i \leq n$; o somatório da aptidão dos indivíduos da população $S_f = \sum_{k=1}^n f_k$. Em seguida é feita uma distribuição normal $d \leftarrow N(0, S_f)$ para determinar qual indivíduo será selecionado. Por fim, é feita uma roleta para determinar qual indivíduo será selecionado, conforme algoritmo 5.4:

Algoritmo 5.4: Roleta

Entrada: População de μ indivíduos; e valor d de escolha da roleta

Saída: I_i , o indivíduo i da população

```
1  $i \leftarrow 1$ , denota o indivíduo  $i$  da população,  $1 \leq i \leq \mu$ 
2  $soma \leftarrow f_i$ 
3 while  $soma < d$  do
4    $i \leftarrow i + 1$ , passa para o próximo indivíduo
5    $soma \leftarrow soma + f_i$ 
6 end
7 Retorna  $I_i$ , o indivíduo  $i$  da população
```

seleção por torneio a seleção por torneio classifica n indivíduos da população, aleatoriamente, para participar do torneio e escolhe um dentre os escolhidos. Para efeito de análises e após alguns testes para afinação da solução, o valor de n foi fixado em 5.

Resultados obtidos

Com o fim de analisar a solução algorítmica proposta, foram criados 27 casos de experimentos realizados sobre 18 instâncias da OR-Library [5]. Cada instância foi exercitada 20 vezes em cada uma das soluções propostas, em cada um dos 27 casos de experimentos. A execução dos casos de experimentos foi feita sobre um Intel Core 2 Duo de 2GHz, 3GB de RAM e sistema operacional GNU/Linux Ubuntu 11.10, Kernel Linux 3.0.0-14-generic. A implementação das soluções foi feita em linguagem *C* — um framework para execução automática dos casos de experimentos foi escrito em *Python*. Conquanto a OR-Library forneça 82 instâncias de problemas de escalonamento job-shop, não necessariamente a literatura realiza o exercício de todas estas instâncias [16] [22] [13] [32].

Os 27 casos de experimentos são organizados conforme delineado na tabela 6.1. Nessa, *IPXO* e *NPXO* referem-se, respectivamente, a recombinação de 1 e n -pontos; *UXO* refere-se à recombinação uniforme; *MP* refere-se a mutação por permutações; *GA* refere-se à geração aleatória de descendentes; *RANK*, *FP* e *T* referem-se, respectivamente, à seleção por ranqueamento linear, proporcional à aptidão e por torneio; e *EXP* refere-se ao número do experimento. Em seguida são apresentadas as 27 tabelas com os resultados obtidos para cada experimento.

Cada tabela com os resultados (da tabela 6.2 à tabela 6.28) apresenta, para cada algoritmo desenvolvido, o melhor resultado obtido nas 20 execuções, a média dos melhores valores obtidos nas 20 execuções, o pior resultado obtido nas 20 execuções e a média populacional de todos os indivíduos que foram criados nas 20 execuções. *GA/EV-JB* refere-se à solução algorítmica baseada na versão canônica dos algoritmos genéticos e *IVF/EV-JB* refere-se à solução híbrida, objeto de desenvolvimento e análise no presente trabalho — conforme descrito no Capítulo 5. *BKS* (*best known solution*) refere-se à melhor solução reconhecida pela literatura para aquela instância [13] [22] [16]. O tamanho ou dimensão da instância é determinado pelo número n de tarefas e o número m de máquinas.

Tabela 6.1: *Configuração dos casos de experimentos*

Configurações								EXP
1PXO	NPXO	UXO	MP	GA	RANK	FP	T	
•			•		•			1
	•		•		•			2
		•	•		•			3
•			•			•		4
	•		•			•		5
		•	•			•		6
•			•				•	7
	•		•				•	8
		•	•				•	9
•				•	•			10
	•			•	•			11
		•		•	•			12
•				•		•		13
	•			•		•		14
		•		•		•		15
•				•			•	16
	•			•			•	17
		•		•			•	18
•			•	•	•			19
	•		•	•	•			20
		•	•	•	•			21
•			•	•		•		22
	•		•	•		•		23
		•	•	•		•		24
•			•	•			•	25
	•		•	•			•	26
		•	•	•			•	27

Tabela 6.2: Resultados do caso de experimento 1

Instância	Tamanho ($n \times m$)	BKS	GA/EV-JB				IVF/EV-JB			
			Melhor	Média	Pior	Média (pop.)	Melhor	Média	Pior	Média (pop.)
FT06	6×6	55	55	55	147	69	55	55	146	69
FT10	10×10	930	961	1018	2498	1243	969	1001	2445	1374
FT20	20×5	1165	1180	1254	2982	1462	1185	1253	2776	1473
LA01	10×5	666	666	671	1737	823	666	666	1699	799
LA02	10×5	655	655	677	1651	878	655	672	1561	849
LA03	10×5	597	603	618	1596	802	605	621	1450	765
LA04	10×5	590	590	609	1607	841	590	606	1512	738
LA05	10×5	593	593	593	1417	698	593	593	1479	698
LA06	15×5	926	926	926	2078	1118	926	926	2080	1110
LA07	15×5	890	890	890	2124	1110	890	890	1986	1111
LA08	15×5	863	863	863	2083	1138	863	863	2123	1066
LA09	15×5	951	951	951	2342	1114	951	951	2120	1216
LA10	15×5	958	958	958	2065	1194	958	958	2029	1186
LA11	20×5	1222	1222	1222	2605	1441	1222	1222	2621	1515
LA12	20×5	1039	1039	1039	2308	1216	1039	1039	2189	1182
LA13	20×5	1150	1150	1150	2497	1317	1150	1150	2374	1313
LA14	20×5	1292	1292	1292	2584	1428	1292	1292	2553	1450
LA15	20×5	1207	1207	1207	2810	1464	1207	1209	2655	1449

Tabela 6.3: Resultados do caso de experimento 2

Instância	Tamanho ($n \times m$)	BKS	GA/EV-JB				IVF/EV-JB			
			Melhor	Média (melhores)	Pior	Média (pop.)	Melhor	Média (melhores)	Pior	Média (pop.)
FT06	6×6	55	55	55	155	77	55	55	139	77
FT10	10×10	930	986	1033	2548	1408	977	1050	2461	1470
FT20	20×5	1165	1202	1256	3060	1724	1196	1267	2817	1608
LA01	10×5	666	666	671	1824	861	666	668	1675	906
LA02	10×5	655	655	676	1626	901	663	667	1640	911
LA03	10×5	597	611	622	1550	855	606	616	1508	796
LA04	10×5	590	595	607	1616	881	593	601	1574	843
LA05	10×5	593	593	593	1445	792	593	593	1420	798
LA06	15×5	926	926	926	2253	1162	926	926	1972	1219
LA07	15×5	890	890	892	2089	1171	890	893	1955	1192
LA08	15×5	863	863	863	2085	1180	863	863	2087	1178
LA09	15×5	951	951	951	2248	1305	951	951	2135	1249
LA10	15×5	958	958	958	2281	1209	958	958	2065	1222
LA11	20×5	1222	1222	1222	2599	1531	1222	1222	2487	1452
LA12	20×5	1039	1039	1039	2341	1342	1039	1039	2351	1417
LA13	20×5	1150	1150	1150	2556	1487	1150	1150	2457	1390
LA14	20×5	1292	1292	1292	2691	1642	1292	1292	2565	1561
LA15	20×5	1207	1207	1223	2806	1689	1207	1207	2769	1482

Tabela 6.4: Resultados do caso de experimento 3

Instância	Tamanho ($n \times m$)	BKS	GA/EV-JB				IVF/EV-JB			
			Melhor	Média	Pior	Média (pop.)	Melhor	Média	Pior	Média (pop.)
FT06	6×6	55	55	55	152	73	55	55	139	78
FT10	10×10	930	996	1047	2554	1603	972	1065	2454	1546
FT20	20×5	1165	1246	1315	2996	1990	1220	1351	2766	1735
LA01	10×5	666	666	667	1713	959	666	669	1663	903
LA02	10×5	655	655	684	1670	974	663	687	1752	924
LA03	10×5	597	609	645	1590	904	617	626	1468	850
LA04	10×5	590	590	597	1661	930	590	604	1518	806
LA05	10×5	593	593	593	1548	725	593	593	1391	803
LA06	15×5	926	926	926	2180	1259	926	926	2027	1195
LA07	15×5	890	890	891	2109	1330	890	890	1957	1234
LA08	15×5	863	863	863	2209	1268	863	863	2044	1250
LA09	15×5	951	951	951	2304	1320	951	951	2145	1211
LA10	15×5	958	958	958	2290	1287	958	958	2051	1248
LA11	20×5	1222	1222	1222	2733	1614	1222	1222	2458	1557
LA12	20×5	1039	1039	1039	2328	1483	1039	1039	2260	1447
LA13	20×5	1150	1150	1150	2611	1571	1150	1150	2450	1612
LA14	20×5	1292	1292	1292	2663	1631	1292	1292	2437	1582
LA15	20×5	1207	1207	1246	2906	1774	1207	1207	2691	1629

Tabela 6.5: Resultados do caso de experimento 4

Instância	Tamanho ($n \times m$)	BKS	GA/EV-JB				IVF/EV-JB			
			Melhor	Média (melhores)	Pior	Média (pop.)	Melhor	Média (melhores)	Pior	Média (pop.)
FT06	6×6	55	55	57	156	87	55	56	140	87
FT10	10×10	930	1156	1185	2630	1716	1143	1171	2511	1722
FT20	20×5	1165	1497	1525	3090	2088	1435	1476	2848	2078
LA01	10×5	666	691	705	1850	1055	669	713	1669	1046
LA02	10×5	655	709	744	1814	1089	710	746	1621	1066
LA03	10×5	597	655	682	1656	996	643	684	1454	984
LA04	10×5	590	640	658	1669	1007	610	675	1565	1001
LA05	10×5	593	593	593	1519	891	593	593	1380	872
LA06	15×5	926	926	944	2439	1369	926	953	2019	1366
LA07	15×5	890	933	971	2222	1412	951	982	1992	1404
LA08	15×5	863	923	943	2361	1381	885	942	2035	1366
LA09	15×5	951	955	994	2391	1429	951	979	2160	1426
LA10	15×5	958	958	959	2233	1357	958	958	2023	1337
LA11	20×5	1222	1259	1303	2764	1753	1233	1286	2471	1750
LA12	20×5	1039	1075	1087	2515	1550	1065	1072	2248	1547
LA13	20×5	1150	1187	1214	2771	1700	1151	1217	2437	1696
LA14	20×5	1292	1292	1292	2802	1739	1292	1292	2576	1737
LA15	20×5	1207	1347	1388	2928	1890	1347	1404	2653	1891

Tabela 6.6: Resultados do caso de experimento 5

Instância	Tamanho ($n \times m$)	BKS	GA/EV-JB				IVF/EV-JB			
			Melhor	Média (melhores)	Pior	Média (pop.)	Melhor	Média (melhores)	Pior	Média (pop.)
FT06	6×6	55	55	57	160	88	55	55	139	87
FT10	10×10	930	1156	1227	2681	1721	1156	1225	2405	1714
FT20	20×5	1165	1447	1489	3176	2088	1467	1501	2881	2070
LA01	10×5	666	678	711	1856	1053	681	701	1781	1048
LA02	10×5	655	727	745	1802	1082	718	737	1587	1071
LA03	10×5	597	662	680	1640	996	648	678	1475	987
LA04	10×5	590	632	665	1707	1009	634	658	1524	992
LA05	10×5	593	593	593	1585	887	593	593	1365	865
LA06	15×5	926	933	965	2318	1370	927	946	2037	1359
LA07	15×5	890	980	1006	2181	1409	973	1000	2056	1405
LA08	15×5	863	920	967	2250	1375	917	947	2087	1364
LA09	15×5	951	972	982	2345	1432	958	973	2196	1422
LA10	15×5	958	958	958	2263	1349	958	958	2039	1338
LA11	20×5	1222	1267	1282	2750	1753	1255	1286	2484	1746
LA12	20×5	1039	1072	1093	2508	1551	1059	1086	2370	1547
LA13	20×5	1150	1164	1205	2723	1704	1182	1203	2449	1690
LA14	20×5	1292	1292	1292	2745	1737	1292	1292	2529	1739
LA15	20×5	1207	1353	1416	2836	1893	1368	1396	2730	1885

Tabela 6.7: Resultados do caso de experimento 6

Instância	Tamanho ($n \times m$)	BKS	GA/EV-JB				IVF/EV-JB			
			Melhor	Média (melhores)	Pior	Média (pop.)	Melhor	Média (melhores)	Pior	Média (pop.)
FT06	6×6	55	55	56	160	90	55	56	136	85
FT10	10×10	930	1162	1229	2713	1719	1148	1210	2456	1710
FT20	20×5	1165	1517	1555	3066	2081	1507	1535	2935	2073
LA01	10×5	666	687	718	1821	1058	681	692	1642	1050
LA02	10×5	655	724	754	1755	1084	714	747	1596	1067
LA03	10×5	597	668	688	1703	995	654	678	1483	988
LA04	10×5	590	651	674	1721	1005	627	660	1551	992
LA05	10×5	593	593	593	1560	888	593	593	1376	876
LA06	15×5	926	927	949	2244	1365	928	943	2178	1359
LA07	15×5	890	957	981	2311	1420	973	997	2200	1395
LA08	15×5	863	931	946	2255	1369	913	960	2017	1366
LA09	15×5	951	951	984	2418	1439	951	983	2164	1415
LA10	15×5	958	958	960	2205	1346	958	958	1988	1340
LA11	20×5	1222	1269	1304	2790	1762	1231	1283	2505	1747
LA12	20×5	1039	1081	1094	2658	1548	1047	1074	2264	1544
LA13	20×5	1150	1188	1218	2792	1701	1189	1207	2448	1696
LA14	20×5	1292	1292	1292	2772	1735	1292	1292	2559	1731
LA15	20×5	1207	1379	1420	2816	1887	1349	1383	2793	1887

Tabela 6.8: Resultados do caso de experimento 7

Instância	Tamanho ($n \times m$)	BKS	GA/EV-JB				IVF/EV-JB			
			Melhor	Média	Pior	Média (pop.)	Melhor	Média	Pior	Média (pop.)
FT06	6×6	55	55	55	154	79	55	55	133	79
FT10	10×10	930	1005	1053	2638	1642	980	1020	2407	1580
FT20	20×5	1165	1303	1347	3053	1948	1252	1267	2765	1817
LA01	10×5	666	666	666	1728	967	666	671	1585	927
LA02	10×5	655	657	693	1703	1006	658	671	1590	920
LA03	10×5	597	619	627	1632	900	604	628	1416	873
LA04	10×5	590	598	611	1787	914	595	610	1491	880
LA05	10×5	593	593	593	1540	779	593	593	1347	774
LA06	15×5	926	926	926	2169	1275	926	926	2033	1270
LA07	15×5	890	890	912	2161	1271	890	890	1899	1250
LA08	15×5	863	863	868	2342	1266	863	863	2008	1235
LA09	15×5	951	951	951	2314	1358	951	951	2100	1283
LA10	15×5	958	958	958	2238	1258	958	958	1943	1235
LA11	20×5	1222	1222	1222	2796	1674	1222	1222	2571	1644
LA12	20×5	1039	1039	1039	2374	1469	1039	1039	2152	1413
LA13	20×5	1150	1150	1150	2574	1607	1150	1150	2384	1586
LA14	20×5	1292	1292	1292	2791	1673	1292	1292	2433	1635
LA15	20×5	1207	1231	1297	2855	1813	1209	1239	2703	1784

Tabela 6.9: Resultados do caso de experimento 8

Instância	Tamanho ($n \times m$)	BKS	GA/EV-JB				IVF/EV-JB			
			Melhor	Média (melhores)	Pior	Média (pop.)	Melhor	Média (melhores)	Pior	Média (pop.)
FT06	6×6	55	55	55	166	85	55	55	138	83
FT10	10×10	930	1131	1167	2706	1690	1059	1115	2422	1693
FT20	20×5	1165	1356	1459	3106	2004	1299	1413	2975	2026
LA01	10×5	666	666	685	1839	1017	666	675	1569	988
LA02	10×5	655	691	727	1764	1061	684	688	1540	1034
LA03	10×5	597	626	654	1716	965	632	649	1498	957
LA04	10×5	590	613	624	1710	932	607	624	1525	957
LA05	10×5	593	593	593	1518	724	593	593	1380	831
LA06	15×5	926	926	933	2203	1336	926	926	1985	1315
LA07	15×5	890	919	944	2166	1360	892	913	2004	1347
LA08	15×5	863	890	927	2230	1343	863	893	1992	1334
LA09	15×5	951	951	962	2360	1391	951	951	2109	1372
LA10	15×5	958	958	958	2199	1305	958	958	2060	1315
LA11	20×5	1222	1238	1280	2770	1736	1222	1242	2517	1729
LA12	20×5	1039	1041	1088	2407	1518	1039	1058	2312	1503
LA13	20×5	1150	1150	1190	2740	1677	1150	1182	2444	1664
LA14	20×5	1292	1292	1292	2842	1689	1292	1292	2534	1712
LA15	20×5	1207	1302	1357	2783	1840	1263	1322	2667	1858

Tabela 6.10: Resultados do caso de experimento 9

Instância	Tamanho ($n \times m$)	BKS	GA/EV-JB				IVF/EV-JB			
			Melhor	Média (melhores)	Pior	Média (pop.)	Melhor	Média (melhores)	Pior	Média (pop.)
FT06	6×6	55	55	55	155	86	55	56	135	83
FT10	10×10	930	1102	1188	2658	1701	1063	1151	2399	1698
FT20	20×5	1165	1413	1480	3129	2033	1342	1437	2784	2022
LA01	10×5	666	666	667	1823	989	666	666	1622	1007
LA02	10×5	655	704	734	1740	1061	672	704	1566	1035
LA03	10×5	597	638	676	1633	974	617	644	1612	963
LA04	10×5	590	624	635	1703	952	607	621	1565	930
LA05	10×5	593	593	593	1549	802	593	593	1416	762
LA06	15×5	926	926	936	2279	1325	926	934	1985	1333
LA07	15×5	890	919	946	2201	1368	894	927	2046	1356
LA08	15×5	863	875	905	2271	1347	863	880	2013	1334
LA09	15×5	951	951	968	2346	1387	951	956	2076	1373
LA10	15×5	958	958	958	2246	1225	958	958	2026	1282
LA11	20×5	1222	1231	1271	2930	1740	1222	1235	2479	1720
LA12	20×5	1039	1039	1061	2443	1526	1039	1041	2245	1514
LA13	20×5	1150	1150	1163	2705	1668	1150	1157	2405	1666
LA14	20×5	1292	1292	1292	2920	1725	1292	1292	2449	1713
LA15	20×5	1207	1311	1386	2987	1867	1308	1352	2672	1849

Tabela 6.11: Resultados do caso de experimento 10

Instância	Tamanho ($n \times m$)	BKS	GA/EV-JB				IVF/EV-JB			
			Melhor	Média (melhores)	Pior	Média (pop.)	Melhor	Média (melhores)	Pior	Média (pop.)
FT06	6×6	55	57	58	149	63	57	57	136	62
FT10	10×10	930	1204	1258	2633	1351	1179	1250	2473	1337
FT20	20×5	1165	1547	1566	3343	1670	1482	1554	2869	1653
LA01	10×5	666	715	728	1753	795	706	745	1543	805
LA02	10×5	655	747	773	1714	835	757	764	1567	821
LA03	10×5	597	680	685	1612	747	675	708	1440	762
LA04	10×5	590	632	697	1717	760	661	689	1457	747
LA05	10×5	593	593	593	1484	648	593	593	1365	641
LA06	15×5	926	937	968	2296	1046	940	975	2086	1046
LA07	15×5	890	999	1024	2257	1100	993	1024	1984	1092
LA08	15×5	863	947	972	2207	1052	938	969	2000	1043
LA09	15×5	951	960	1008	2331	1090	991	1005	2116	1084
LA10	15×5	958	958	964	2360	1044	958	978	1956	1049
LA11	20×5	1222	1276	1322	2730	1410	1257	1320	2435	1399
LA12	20×5	1039	1098	1127	2497	1212	1063	1110	2225	1191
LA13	20×5	1150	1219	1247	2641	1338	1210	1256	2455	1340
LA14	20×5	1292	1292	1306	2698	1394	1292	1292	2488	1377
LA15	20×5	1207	1340	1419	2758	1512	1416	1450	2735	1537

Tabela 6.12: Resultados do caso de experimento 11

Instância	Tamanho ($n \times m$)	BKS	GA/EV-JB				IVF/EV-JB			
			Melhor	Média (melhores)	Pior	Média (pop.)	Melhor	Média (melhores)	Pior	Média (pop.)
FT06	6×6	55	58	58	153	63	56	58	133	63
FT10	10×10	930	1162	1247	2632	1344	1204	1249	2353	1338
FT20	20×5	1165	1538	1558	3000	1668	1511	1563	2783	1660
LA01	10×5	666	690	736	1744	801	702	738	1557	798
LA02	10×5	655	743	781	1698	842	739	766	1563	822
LA03	10×5	597	672	688	1602	749	679	700	1459	753
LA04	10×5	590	640	688	1616	754	663	692	1490	752
LA05	10×5	593	593	593	1479	648	593	594	1312	644
LA06	15×5	926	935	962	2127	1037	926	977	1936	1046
LA07	15×5	890	998	1027	2145	1103	962	1027	1998	1098
LA08	15×5	863	909	958	2269	1041	915	968	2005	1047
LA09	15×5	951	972	997	2273	1085	965	996	2065	1076
LA10	15×5	958	958	985	2212	1060	958	972	1924	1044
LA11	20×5	1222	1275	1299	2727	1387	1258	1289	2413	1376
LA12	20×5	1039	1061	1079	2436	1172	1085	1121	2172	1204
LA13	20×5	1150	1203	1257	2660	1344	1202	1249	2412	1333
LA14	20×5	1292	1292	1303	2722	1398	1292	1292	2458	1378
LA15	20×5	1207	1397	1425	2853	1519	1385	1409	2704	1502

Tabela 6.13: Resultados do caso de experimento 12

Instância	Tamanho ($n \times m$)	BKS	GA/EV-JB				IVF/EV-JB			
			Melhor	Média	Pior	Média (pop.)	Melhor	Média	Pior	Média (pop.)
FT06	6×6	55	57	57	148	63	55	59	145	64
FT10	10×10	930	1190	1265	2561	1357	1182	1210	2467	1307
FT20	20×5	1165	1514	1566	2978	1668	1504	1560	2859	1660
LA01	10×5	666	675	730	1719	797	675	704	1606	771
LA02	10×5	655	751	765	1708	826	720	780	1571	835
LA03	10×5	597	674	683	1608	743	683	698	1434	749
LA04	10×5	590	653	695	1662	759	653	689	1459	748
LA05	10×5	593	593	593	1516	649	593	593	1346	644
LA06	15×5	926	940	973	2222	1053	934	973	1916	1045
LA07	15×5	890	978	1014	2183	1090	986	1023	1938	1096
LA08	15×5	863	950	975	2129	1057	906	961	1937	1037
LA09	15×5	951	957	999	2250	1083	959	1023	2091	1100
LA10	15×5	958	958	968	2208	1049	958	977	1935	1047
LA11	20×5	1222	1283	1309	2807	1395	1275	1296	2379	1381
LA12	20×5	1039	1067	1130	2446	1213	1083	1117	2216	1200
LA13	20×5	1150	1189	1238	2618	1328	1213	1249	2336	1334
LA14	20×5	1292	1292	1295	2765	1384	1292	1294	2420	1378
LA15	20×5	1207	1386	1428	2794	1519	1399	1442	2597	1525

Tabela 6.14: Resultados do caso de experimento 13

Instância	Tamanho ($n \times m$)	BKS	GA/EV-JB				IVF/EV-JB			
			Melhor	Média (melhores)	Pior	Média (pop.)	Melhor	Média (melhores)	Pior	Média (pop.)
FT06	6×6	55	55	57	145	84	55	56	145	82
FT10	10×10	930	1172	1216	2640	1672	1164	1200	2331	1670
FT20	20×5	1165	1481	1543	3037	2026	1470	1518	2841	2025
LA01	10×5	666	668	709	1811	1021	687	707	1660	1023
LA02	10×5	655	727	763	1746	1035	728	764	1740	1040
LA03	10×5	597	668	690	1555	964	655	678	1482	954
LA04	10×5	590	648	670	1638	983	632	666	1493	970
LA05	10×5	593	593	593	1448	831	593	593	1318	831
LA06	15×5	926	936	954	2106	1320	928	956	2054	1305
LA07	15×5	890	972	995	2183	1363	951	997	2019	1355
LA08	15×5	863	928	947	2165	1329	906	944	2018	1322
LA09	15×5	951	951	989	2347	1384	957	972	2111	1377
LA10	15×5	958	958	961	2197	1321	958	959	1954	1308
LA11	20×5	1222	1269	1290	2637	1700	1252	1270	2442	1697
LA12	20×5	1039	1077	1101	2359	1504	1052	1072	2185	1493
LA13	20×5	1150	1197	1215	2643	1644	1179	1192	2353	1648
LA14	20×5	1292	1292	1292	2744	1692	1292	1292	2419	1684
LA15	20×5	1207	1371	1390	2805	1847	1359	1407	2759	1839

Tabela 6.15: Resultados do caso de experimento 14

Instância	Tamanho ($n \times m$)	BKS	GA/EV-JB				IVF/EV-JB			
			Melhor	Média (melhores)	Pior	Média (pop.)	Melhor	Média (melhores)	Pior	Média (pop.)
FT06	6×6	55	55	57	150	84	55	55	135	83
FT10	10×10	930	1130	1213	2572	1678	1173	1217	2520	1669
FT20	20×5	1165	1493	1544	3063	2030	1495	1518	2829	2034
LA01	10×5	666	689	710	1748	1017	680	693	1573	1016
LA02	10×5	655	725	748	1697	1039	732	752	1646	1039
LA03	10×5	597	670	683	1597	956	662	689	1395	953
LA04	10×5	590	639	675	1618	982	637	674	1455	972
LA05	10×5	593	593	593	1559	840	593	593	1300	828
LA06	15×5	926	939	961	2220	1315	926	948	2010	1303
LA07	15×5	890	956	978	2165	1361	955	974	2108	1353
LA08	15×5	863	908	931	2184	1330	927	935	1981	1331
LA09	15×5	951	974	982	2347	1375	954	980	2175	1375
LA10	15×5	958	958	958	2172	1312	958	960	1960	1311
LA11	20×5	1222	1266	1285	2622	1707	1247	1281	2536	1695
LA12	20×5	1039	1055	1094	2426	1500	1049	1100	2203	1497
LA13	20×5	1150	1196	1232	2602	1639	1159	1205	2566	1649
LA14	20×5	1292	1292	1292	2565	1697	1292	1292	2442	1688
LA15	20×5	1207	1342	1411	3031	1841	1369	1421	2531	1843

Tabela 6.16: Resultados do caso de experimento 15

Instância	Tamanho ($n \times m$)	BKS	GA/EV-JB				IVF/EV-JB			
			Melhor	Média (melhores)	Pior	Média (pop.)	Melhor	Média (melhores)	Pior	Média (pop.)
FT06	6×6	55	56	57	148	85	55	56	134	83
FT10	10×10	930	1174	1211	2582	1676	1145	1202	2399	1664
FT20	20×5	1165	1478	1554	2985	2032	1494	1524	2807	2025
LA01	10×5	666	678	709	1771	1024	677	709	1600	1020
LA02	10×5	655	735	752	1754	1035	718	760	1539	1026
LA03	10×5	597	664	691	1564	961	666	679	1419	959
LA04	10×5	590	649	664	1613	979	619	663	1566	973
LA05	10×5	593	593	593	1476	830	593	593	1359	828
LA06	15×5	926	926	946	2290	1311	929	945	1940	1305
LA07	15×5	890	975	983	2088	1364	968	999	1972	1350
LA08	15×5	863	919	938	2336	1329	891	940	1971	1319
LA09	15×5	951	957	983	2197	1387	967	987	2012	1368
LA10	15×5	958	958	959	2272	1317	958	958	1974	1312
LA11	20×5	1222	1259	1286	2760	1699	1258	1279	2394	1702
LA12	20×5	1039	1072	1091	2387	1498	1041	1103	2154	1499
LA13	20×5	1150	1193	1202	2657	1651	1187	1210	2382	1646
LA14	20×5	1292	1292	1292	2613	1699	1292	1292	2452	1690
LA15	20×5	1207	1391	1406	2806	1842	1391	1430	2796	1841

Tabela 6.17: Resultados do caso de experimento 16

Instância	Tamanho ($n \times m$)	BKS	GA/EV-JB				IVF/EV-JB			
			Melhor	Média (melhores)	Pior	Média (pop.)	Melhor	Média (melhores)	Pior	Média (pop.)
FT06	6×6	55	55	58	151	63	55	58	138	63
FT10	10×10	930	1186	1225	2609	1342	1187	1241	2340	1332
FT20	20×5	1165	1517	1545	2990	1674	1495	1564	2845	1662
LA01	10×5	666	692	715	1739	785	696	729	1510	788
LA02	10×5	655	752	766	1742	829	713	769	1529	826
LA03	10×5	597	654	695	1565	755	677	704	1466	754
LA04	10×5	590	661	693	1705	758	647	670	1577	735
LA05	10×5	593	593	594	1508	648	593	593	1346	644
LA06	15×5	926	926	976	2167	1059	951	982	1902	1057
LA07	15×5	890	935	1023	2104	1100	974	1019	1983	1092
LA08	15×5	863	920	964	2194	1051	938	957	1931	1035
LA09	15×5	951	986	1014	2296	1097	971	1014	2110	1094
LA10	15×5	958	958	977	2159	1056	958	965	2066	1039
LA11	20×5	1222	1260	1312	2717	1411	1269	1290	2420	1378
LA12	20×5	1039	1070	1121	2450	1211	1076	1125	2205	1205
LA13	20×5	1150	1212	1244	2651	1334	1210	1250	2340	1336
LA14	20×5	1292	1292	1292	2672	1382	1292	1292	2437	1377
LA15	20×5	1207	1391	1461	2768	1546	1378	1459	2570	1544

Tabela 6.18: Resultados do caso de experimento 17

Instância	Tamanho ($n \times m$)	BKS	GA/EV-JB				IVF/EV-JB			
			Melhor	Média (melhores)	Pior	Média (pop.)	Melhor	Média (melhores)	Pior	Média (pop.)
FT06	6×6	55	55	58	159	64	55	58	136	64
FT10	10×10	930	1183	1236	2559	1424	1168	1250	2431	1342
FT20	20×5	1165	1472	1513	3038	1623	1515	1560	2806	1658
LA01	10×5	666	701	722	1786	798	699	732	1571	793
LA02	10×5	655	737	772	1766	829	722	773	1703	830
LA03	10×5	597	672	704	1564	762	663	700	1442	755
LA04	10×5	590	637	683	1639	750	667	691	1485	750
LA05	10×5	593	593	594	1614	647	593	594	1355	643
LA06	15×5	926	931	974	2188	1057	941	955	1921	1031
LA07	15×5	890	978	1007	2166	1126	980	1015	1957	1090
LA08	15×5	863	917	979	2332	1064	921	954	2005	1031
LA09	15×5	951	975	1016	2437	1097	961	1014	2049	1094
LA10	15×5	958	958	965	2162	1046	958	982	1986	1051
LA11	20×5	1222	1255	1295	2653	1391	1247	1292	2469	1379
LA12	20×5	1039	1074	1104	2478	1207	1081	1088	2165	1175
LA13	20×5	1150	1200	1234	2711	1328	1196	1240	2479	1327
LA14	20×5	1292	1292	1293	2686	1384	1292	1292	2419	1377
LA15	20×5	1207	1407	1433	2951	1528	1375	1427	2602	1516

Tabela 6.19: Resultados do caso de experimento 18

Instância	Tamanho ($n \times m$)	BKS	GA/EV-JB				IVF/EV-JB			
			Melhor	Média (melhores)	Pior	Média (pop.)	Melhor	Média (melhores)	Pior	Média (pop.)
FT06	6×6	55	55	57	158	63	57	58	137	62
FT10	10×10	930	1169	1236	2655	1350	1185	1244	2414	1334
FT20	20×5	1165	1457	1514	3017	1975	1469	1567	2930	1665
LA01	10×5	666	686	717	1807	787	694	710	1601	776
LA02	10×5	655	736	783	1814	840	750	777	1552	834
LA03	10×5	597	682	706	1588	762	679	700	1402	753
LA04	10×5	590	660	702	1660	765	648	688	1486	751
LA05	10×5	593	593	594	1499	650	593	593	1282	643
LA06	15×5	926	926	956	2320	1035	931	967	2014	1041
LA07	15×5	890	976	1002	2216	1084	982	998	1974	1074
LA08	15×5	863	936	955	2250	1039	936	965	2000	1042
LA09	15×5	951	966	987	2269	1073	951	1002	2060	1081
LA10	15×5	958	958	979	2194	1061	958	964	1926	1039
LA11	20×5	1222	1266	1301	2781	1621	1263	1298	2437	1383
LA12	20×5	1039	1085	1102	2420	1207	1085	1113	2164	1196
LA13	20×5	1150	1188	1222	2604	1448	1197	1224	2323	1313
LA14	20×5	1292	1292	1292	2723	1600	1292	1292	2486	1378
LA15	20×5	1207	1383	1419	2855	1645	1388	1433	2617	1516

Tabela 6.20: Resultados do caso de experimento 19

Instância	Tamanho ($n \times m$)	BKS	GA/EV-JB				IVF/EV-JB			
			Melhor	Média	Pior	Média (pop.)	Melhor	Média	Pior	Média (pop.)
FT06	6×6	55	55	55	148	70	55	56	134	69
FT10	10×10	930	963	1047	2684	1435	967	992	2356	1262
FT20	20×5	1165	1180	1255	3417	1603	1218	1251	2839	1516
LA01	10×5	666	666	666	1705	814	666	679	1636	841
LA02	10×5	655	655	666	1805	865	655	686	1554	872
LA03	10×5	597	603	617	1655	782	597	623	1455	783
LA04	10×5	590	590	600	1650	767	590	606	1484	776
LA05	10×5	593	593	593	1447	719	593	593	1410	719
LA06	15×5	926	926	926	2120	1118	926	926	1984	1085
LA07	15×5	890	890	890	2110	1073	890	890	1930	1094
LA08	15×5	863	863	863	2138	1124	863	863	2011	1047
LA09	15×5	951	951	951	2299	1128	951	951	2089	1133
LA10	15×5	958	958	958	2261	1116	958	958	2085	1142
LA11	20×5	1222	1222	1222	2695	1402	1222	1222	2386	1453
LA12	20×5	1039	1039	1039	2464	1237	1039	1039	2226	1263
LA13	20×5	1150	1150	1150	2576	1376	1150	1150	2455	1385
LA14	20×5	1292	1292	1292	2697	1489	1292	1292	2517	1474
LA15	20×5	1207	1207	1207	2842	1503	1207	1211	2529	1467

Tabela 6.21: Resultados do caso de experimento 20

Instância	Tamanho ($n \times m$)	BKS	GA/EV-JB				IVF/EV-JB			
			Melhor	Média	Pior	Média (pop.)	Melhor	Média	Pior	Média (pop.)
FT06	6×6	55	55	55	152	71	55	55	142	82
FT10	10×10	930	983	1037	2610	1459	982	1058	2359	1566
FT20	20×5	1165	1185	1301	3028	1765	1240	1338	2903	1881
LA01	10×5	666	666	677	1731	944	666	671	1563	965
LA02	10×5	655	655	668	1696	868	676	722	1578	987
LA03	10×5	597	597	634	1581	854	613	627	1473	907
LA04	10×5	590	590	606	1695	808	598	614	1428	939
LA05	10×5	593	593	593	1554	747	593	593	1353	802
LA06	15×5	926	926	926	2151	1193	926	926	1942	1261
LA07	15×5	890	890	897	2245	1173	890	924	2046	1290
LA08	15×5	863	863	863	2259	1203	863	875	1925	1269
LA09	15×5	951	951	951	2266	1225	951	951	2058	1338
LA10	15×5	958	958	958	2210	1134	958	958	2003	1268
LA11	20×5	1222	1222	1222	2755	1592	1222	1222	2449	1590
LA12	20×5	1039	1039	1039	2468	1320	1039	1039	2171	1458
LA13	20×5	1150	1150	1150	2639	1490	1150	1150	2367	1564
LA14	20×5	1292	1292	1292	2668	1543	1292	1292	2421	1644
LA15	20×5	1207	1207	1207	2870	1584	1207	1244	2597	1742

Tabela 6.22: Resultados do caso de experimento 21

Instância	Tamanho ($n \times m$)	BKS	GA/EV-JB				IVF/EV-JB			
			Melhor	Média	Pior	Média (pop.)	Melhor	Média	Pior	Média (pop.)
FT06	6×6	55	55	55	149	77	55	55	132	82
FT10	10×10	930	1002	1046	2586	1623	1023	1189	2405	1654
FT20	20×5	1165	1208	1282	3073	1880	1305	1370	2807	1966
LA01	10×5	666	666	680	1804	958	666	670	1582	979
LA02	10×5	655	665	674	1684	999	665	702	1512	1006
LA03	10×5	597	603	627	1785	907	617	635	1434	939
LA04	10×5	590	590	619	1682	919	607	632	1612	937
LA05	10×5	593	593	593	1512	774	593	593	1350	809
LA06	15×5	926	926	926	2240	1248	926	926	1908	1284
LA07	15×5	890	890	908	2100	1283	890	925	1967	1306
LA08	15×5	863	863	873	2251	1227	863	888	1919	1298
LA09	15×5	951	951	951	2378	1332	951	954	2070	1333
LA10	15×5	958	958	958	2153	1199	958	958	1912	1293
LA11	20×5	1222	1222	1222	2693	1650	1222	1230	2419	1659
LA12	20×5	1039	1039	1039	2479	1432	1039	1053	2203	1467
LA13	20×5	1150	1150	1150	2624	1603	1150	1178	2410	1597
LA14	20×5	1292	1292	1292	2750	1588	1292	1292	2511	1658
LA15	20×5	1207	1207	1241	2829	1778	1244	1312	2568	1829

Tabela 6.23: Resultados do caso de experimento 22

Instância	Tamanho ($n \times m$)	BKS	GA/EV-JB				IVF/EV-JB			
			Melhor	Média (melhores)	Pior	Média (pop.)	Melhor	Média (melhores)	Pior	Média (pop.)
FT06	6×6	55	55	57	149	83	55	56	139	83
FT10	10×10	930	1109	1197	2707	1669	1140	1203	2428	1674
FT20	20×5	1165	1492	1533	3000	2031	1451	1514	2728	2025
LA01	10×5	666	692	703	1731	1034	683	697	1553	1025
LA02	10×5	655	724	753	1760	1038	691	745	1520	1038
LA03	10×5	597	647	667	1632	954	641	663	1423	953
LA04	10×5	590	633	658	1678	974	627	677	1484	975
LA05	10×5	593	593	593	1452	835	593	593	1355	823
LA06	15×5	926	934	953	2121	1314	926	947	1980	1313
LA07	15×5	890	969	1008	2239	1362	960	981	1976	1357
LA08	15×5	863	903	939	2149	1329	892	918	1932	1320
LA09	15×5	951	952	989	2319	1382	951	971	2226	1371
LA10	15×5	958	958	959	2204	1313	958	961	2030	1302
LA11	20×5	1222	1237	1276	2654	1696	1250	1274	2512	1696
LA12	20×5	1039	1064	1082	2356	1495	1056	1064	2207	1491
LA13	20×5	1150	1172	1193	2663	1647	1156	1200	2335	1639
LA14	20×5	1292	1292	1292	2792	1685	1292	1292	2513	1684
LA15	20×5	1207	1365	1410	2942	1841	1338	1367	2584	1837

Tabela 6.24: Resultados do caso de experimento 23

Instância	Tamanho ($n \times m$)	BKS	GA/EV-JB				IVF/EV-JB			
			Melhor	Média (melhores)	Pior	Média (pop.)	Melhor	Média (melhores)	Pior	Média (pop.)
FT06	6×6	55	55	57	150	84	55	56	140	82
FT10	10×10	930	1169	1206	2631	1676	1152	1176	2474	1667
FT20	20×5	1165	1448	1547	3127	2036	1494	1525	2792	2025
LA01	10×5	666	680	718	1898	1019	675	701	1599	1018
LA02	10×5	655	718	741	1701	1038	728	745	1590	1033
LA03	10×5	597	655	688	1629	961	652	685	1435	959
LA04	10×5	590	642	663	1789	972	635	660	1522	967
LA05	10×5	593	593	593	1501	835	593	593	1360	831
LA06	15×5	926	940	954	2151	1317	926	950	2040	1317
LA07	15×5	890	956	988	2274	1358	955	986	1958	1360
LA08	15×5	863	912	939	2361	1327	905	950	1950	1327
LA09	15×5	951	957	993	2338	1381	962	985	2042	1373
LA10	15×5	958	958	965	2200	1311	958	961	2066	1307
LA11	20×5	1222	1243	1285	2653	1703	1245	1286	2379	1700
LA12	20×5	1039	1080	1103	2591	1499	1039	1082	2258	1502
LA13	20×5	1150	1190	1217	2710	1652	1174	1208	2388	1639
LA14	20×5	1292	1292	1293	2861	1691	1292	1292	2524	1682
LA15	20×5	1207	1371	1410	2884	1846	1363	1405	2616	1837

Tabela 6.25: Resultados do caso de experimento 24

Instância	Tamanho ($n \times m$)	BKS	GA/EV-JB				IVF/EV-JB			
			Melhor	Média (melhores)	Pior	Média (pop.)	Melhor	Média (melhores)	Pior	Média (pop.)
FT06	6×6	55	55	57	147	83	55	56	129	83
FT10	10×10	930	1164	1200	2572	1681	1145	1203	2365	1671
FT20	20×5	1165	1509	1539	3195	2028	1502	1539	2859	2023
LA01	10×5	666	678	700	1904	1026	679	710	1528	1016
LA02	10×5	655	730	749	1709	1039	722	756	1559	1035
LA03	10×5	597	660	684	1602	962	662	689	1436	955
LA04	10×5	590	651	676	1640	976	633	676	1461	977
LA05	10×5	593	593	593	1506	836	593	593	1347	831
LA06	15×5	926	928	952	2112	1314	926	939	2063	1311
LA07	15×5	890	968	991	2115	1359	956	989	1962	1357
LA08	15×5	863	920	947	2167	1327	923	949	2016	1322
LA09	15×5	951	956	989	2234	1382	951	986	2012	1373
LA10	15×5	958	958	960	2242	1309	958	958	2040	1306
LA11	20×5	1222	1268	1292	2612	1702	1245	1285	2468	1696
LA12	20×5	1039	1078	1099	2441	1496	1054	1095	2220	1494
LA13	20×5	1150	1170	1202	2620	1650	1159	1221	2307	1640
LA14	20×5	1292	1292	1292	2593	1689	1292	1292	2460	1683
LA15	20×5	1207	1384	1420	2948	1848	1379	1406	2587	1842

Tabela 6.26: Resultados do caso de experimento 25

Instância	Tamanho ($n \times m$)	BKS	GA/EV-JB				IVF/EV-JB			
			Melhor	Média	Pior	Média (pop.)	Melhor	Média	Pior	Média (pop.)
FT06	6×6	55	55	55	151	81	55	55	136	74
FT10	10×10	930	1036	1125	2580	1663	988	1032	2409	1576
FT20	20×5	1165	1255	1367	3201	1910	1261	1285	2927	1817
LA01	10×5	666	666	668	1823	971	666	666	1699	922
LA02	10×5	655	659	673	1789	982	663	681	1515	958
LA03	10×5	597	613	642	1565	890	603	628	1529	856
LA04	10×5	590	593	614	1630	878	590	617	1553	898
LA05	10×5	593	593	593	1476	799	593	593	1335	742
LA06	15×5	926	926	927	2238	1289	926	926	1923	1229
LA07	15×5	890	890	930	2139	1323	890	891	1942	1237
LA08	15×5	863	863	889	2208	1252	863	866	2026	1218
LA09	15×5	951	951	955	2280	1335	951	951	2115	1274
LA10	15×5	958	958	958	2377	1280	958	958	2000	1235
LA11	20×5	1222	1222	1222	2669	1682	1222	1222	2427	1619
LA12	20×5	1039	1039	1043	2432	1482	1039	1039	2187	1424
LA13	20×5	1150	1150	1157	2524	1619	1150	1150	2293	1569
LA14	20×5	1292	1292	1292	2720	1677	1292	1292	2449	1637
LA15	20×5	1207	1255	1295	2779	1821	1207	1242	2583	1731

Tabela 6.27: Resultados do caso de experimento 26

Instância	Tamanho ($n \times m$)	BKS	GA/EV-JB				IVF/EV-JB			
			Melhor	Média	Pior	Média (pop.)	Melhor	Média	Pior	Média (pop.)
FT06	6×6	55	55	55	146	82	55	55	146	80
FT10	10×10	930	1097	1165	2621	1673	1085	1133	2304	1669
FT20	20×5	1165	1431	1515	3042	2018	1343	1407	2939	2001
LA01	10×5	666	666	688	1832	1020	666	674	1571	999
LA02	10×5	655	699	731	1671	1031	669	729	1537	1022
LA03	10×5	597	626	654	1621	955	619	651	1426	940
LA04	10×5	590	614	630	1697	944	598	634	1523	938
LA05	10×5	593	593	593	1497	824	593	593	1456	820
LA06	15×5	926	926	933	2198	1301	926	926	2022	1291
LA07	15×5	890	930	968	2218	1344	893	931	1917	1335
LA08	15×5	863	885	929	2151	1316	863	907	2001	1310
LA09	15×5	951	951	964	2474	1370	951	954	2221	1359
LA10	15×5	958	958	958	2129	1303	958	958	1993	1297
LA11	20×5	1222	1234	1264	2641	1694	1222	1231	2501	1691
LA12	20×5	1039	1039	1075	2335	1487	1039	1059	2194	1487
LA13	20×5	1150	1150	1185	2602	1651	1150	1154	2475	1626
LA14	20×5	1292	1292	1292	2884	1687	1292	1292	2396	1677
LA15	20×5	1207	1315	1348	2762	1838	1298	1357	2652	1821

Tabela 6.28: Resultados do caso de experimento 27

Instância	Tamanho ($n \times m$)	BKS	GA/EV-JB				IVF/EV-JB			
			Melhor	Média (melhores)	Pior	Média (pop.)	Melhor	Média (melhores)	Pior	Média (pop.)
FT06	6×6	55	55	55	145	82	55	55	134	82
FT10	10×10	930	1111	1145	2573	1671	1082	1163	2383	1665
FT20	20×5	1165	1420	1493	3025	2025	1355	1474	2759	2012
LA01	10×5	666	666	693	1790	1021	666	679	1584	1003
LA02	10×5	655	713	749	1715	1029	681	707	1514	1028
LA03	10×5	597	649	673	1645	948	618	646	1448	938
LA04	10×5	590	619	651	1648	959	616	631	1476	940
LA05	10×5	593	593	593	1482	823	593	593	1358	821
LA06	15×5	926	926	938	2228	1307	926	926	1929	1299
LA07	15×5	890	936	952	2096	1349	907	958	1982	1341
LA08	15×5	863	886	915	2182	1318	863	907	1976	1310
LA09	15×5	951	951	970	2299	1373	951	951	2031	1362
LA10	15×5	958	958	958	2079	1307	958	958	2050	1299
LA11	20×5	1222	1230	1273	2631	1703	1222	1238	2461	1690
LA12	20×5	1039	1039	1067	2430	1487	1039	1048	2218	1484
LA13	20×5	1150	1150	1200	2556	1637	1150	1170	2311	1640
LA14	20×5	1292	1292	1292	2795	1691	1292	1292	2425	1680
LA15	20×5	1207	1342	1375	2927	1844	1315	1351	2584	1831

6.1 Efetividade da solução

Pelos 27 casos de experimentos é possível avaliar que ambas as soluções desenvolvidas são soluções de qualidade por apresentarem respostas próximas ou semelhantes àquelas encontradas pela literatura. Embora em alguns casos de experimento as soluções não forneçam a melhor solução conhecida (*BKS*), ainda assim obtemos uma resposta relevante. Nos experimentos 1 (tabela 6.2), 2 (tabela 6.3), 3 (tabela 6.4) e 19 (tabela 6.20) obtemos respostas próximas ou semelhantes à melhor solução conhecida em mais de 85% das instâncias exercitadas. Nos experimentos 6 (tabela 6.7), 7 (tabela 6.8), 8 (tabela 6.9), 9 (tabela 6.10), 10 (tabela 6.11), 13 (tabela 6.14), 22 (tabela 6.23), 24 (tabela 6.25), 25 (tabela 6.26), 26 (tabela 6.27) e 27 (tabela 6.28) o IVF/EV-JB obtém ao menos 80% melhores resultados frente ao GA/EV-JB. Nos experimentos 20 (tabela 6.21) e 21 (tabela 6.22) há uma perda na qualidade das soluções encontradas pelo IVF/EV-JB frente ao GA/EV-JB. Nos experimentos 11 (tabela 6.12), 12 (tabela 6.13), 14 (tabela 6.15), 15 (tabela 6.16), 16 (tabela 6.17), 17 (tabela 6.18), 18 (tabela 6.19) e 23 (tabela 6.24) o IVF/EV-JB perde grande parte da sua efetividade e obtém ao menos 55% melhores resultados frente ao GA/EV-JB.

Analisando os piores resultados desenvolvidos em cada solução, o IVF/EV-JB se mostra uma solução inicialmente gulosa, por ter os menores valores de pior indivíduo gerado nas execuções; contudo, esse comportamento não demonstra ser propício a tornar a solução vulnerável em ótimos locais. A média populacional do IVF/EV-JB juntamente os valores de pior indivíduo corroboram com a motivação do algoritmo auxiliar paralelo (AAP) em manter qualidades genotípicas na população e prover resultados de qualidade.

6.2 Influência dos operadores de variabilidade e de seleção sobre a efetividade da solução

Pelos resultados obtidos não é possível identificar quais são as reais influências dos operadores de variabilidade e seleção sobre a efetividade da solução. Contudo, é possível aferir que as melhores configurações das soluções propostas foram as configurações dos experimentos 1 (tabela 6.2), 2 (tabela 6.3), 3 (tabela 6.4), 25 (tabela 6.26), 26 (tabela 6.27) e 27 (tabela 6.28); as piores configurações estão nos experimentos 16 (tabela 6.17), 17 (tabela 6.18) e 18 (tabela 6.19).

Pelos resultados não é possível aferir qual é o melhor ou pior operador de variabilidade ou de seleção, mas qual a configuração é mais propícia a gerar resultados de qualidade, em consonância com os desenvolvimentos da literatura a respeito dos algoritmos evolucionários e como uma solução metaheurística nestes parâmetros deve ser projetada [20]. É possível identificar que a seleção por ranqueamento linear (com alta

pressão seletiva [20] [10]) provê resultados de qualidade quando combinada à mutação por permutação; e também possível verificar a partir dos experimentos que a seleção por torneio (com média pressão seletiva [20] [10]) provê resultados de qualidade quando combinada à mutação por permutação juntamente com a geração aleatória de indivíduos.

Alguns autores defendem que mecanismos de seleção de alta pressão seletiva devem ser combinados com mecanismos que geram grande variabilidade a fim de obter-se resultados de qualidade [20]. Por esta consideração, será válido afirmar que o mecanismo de mutação por permutação é um operador de variabilidade que gera grande variabilidade em nível fenotípico, tornando os indivíduos–soluções aptos a sofrer maior pressão seletiva. A geração aleatória de indivíduos, conforme demonstrados nos experimentos 25 (tabela 6.26), 26 (tabela 6.27) e 27 (tabela 6.28), aparentemente diminui os efeitos da mutação por permutação e tornam os indivíduos–soluções aptos a sofrer uma menor pressão seletiva.

Conclusão

A partir de uma classe de problemas de otimização — problemas de escalonamento job-shop —, foi realizada a análise do algoritmo auxiliar paralelo (AAP) sobre uma solução baseada em algoritmos genéticos, mas com influências de outras abordagens evolucionárias. Além desta análise central, foi também realizada uma análise dos efeitos que operadores de seleção e de variabilidade (recombinação e mutação) influem sobre as soluções–respostas obtidas por estes algoritmos.

Foi possível constatar a relativa efetividade e eficiência das soluções construídas a partir de 27 casos de experimentos, o que contribui para enaltecer a importância das metaheurísticas, sobretudo dos algoritmos evolucionários, como métodos de solução para problemas em otimização. Os casos de experimentos constatarem que a partir de configurações, análises e ajustes obtêm-se resultados de qualidade com algoritmos conceitualmente simples e de fácil aplicabilidade, mesmo em classes de problemas de difícil modelagem matemática, como é o caso dos problemas de escalonamento job-shop [12].

O AAP se consolida como uma proposta de melhoria não só em algoritmos genéticos, mas, também, em outras classes de algoritmos evolucionários. Obteve-se uma resposta positiva do AAP frente à sua contrapartida nos experimentos. Além disso, constatou-se a importância da escolha dos mecanismos de variabilidade e de seleção em uma solução baseada em algoritmos evolucionários.

7.1 Trabalhos futuros

O presente trabalho realizou a aplicação de uma extensão dos algoritmos genéticos — a saber, o algoritmo auxiliar paralelo (AAP) baseado na fertilização in vitro — como solução para problemas de escalonamento job-shop. Em seguida, foi feito um comparativo com uma versão dita canônica com a finalidade de evidenciar os efeitos do AAP sobre a solução metaheurística e suas influências nas soluções obtidas.

Possibilidades de expansão do presente trabalho seria aplicar o AAP em outras classes de algoritmos evolucionários, como as estratégias evolutivas, ou em outras abordagens metaheurísticas baseadas num conjunto de soluções–indivíduos, ou populações

de indivíduos (“busca adaptativa paralela”, como *particle swarm optimization*). O AAP tem por base conceitos simples, como um indivíduo representado genotípica ou fenotipicamente; mecanismos de geração de novas soluções—indivíduos e de variabilidade destas soluções. Portanto, a aplicabilidade do AAP em outras classes de algoritmos evolucionários garante-se pois este é de fácil implementação, necessitando apenas de conceitos inerentes aos algoritmos evolucionários.

Outra expansão possível é realizar um comparativo mais ostensivo do AAP frente a outras abordagens híbridas, como os *immune genetic algorithms*.

Referências Bibliográficas

- [1] ADIBI, M. A.; ZANDIEH, M.; AMIRI, M. **Multi-objective scheduling of dynamic job shop using variable neighborhood search**. *Expert Systems with Applications*, 37:282–287, 2010.
- [2] ARTIGUES, C.; GENDREAU, M.; ROUSSEAU, L.-M.; VERGNAUD, A. **Solving an integrated employee timetabling and job-shop scheduling problem via hybrid branch-and-bound**. *Computers and Operations Research*, 36:2330–2340, 2009.
- [3] BEAN, J. C. **Genetic algorithms and random keys for sequencing and optimization**. *ORSA Journal on Computing*, 6(2):154–160, 1994.
- [4] BEAN, J. C.; NORMAN, B. A. **Random keys for job shop scheduling**. Technical report, Department of Industrial and Operations Engineering, University of Michigan, January 1993.
- [5] BEASLEY, J. E. **Or-library**. <http://people.brunel.ac.uk/~mastjjb/jeb/info.html>, Dezembro 2011.
- [6] BRUCKER, P. **Scheduling Algorithms**. Springer, Berlin, Deutschland, 2007.
- [7] CAMILO-JUNIOR, C. G. **Algoritmo Auxiliar Paralelo inspirado na Fertilização in Vitro para melhorar o desempenho de Algoritmos Genéticos**. PhD thesis, Universidade Federal de Uberlândia, Março 2010.
- [8] CAMILO-JUNIOR, C. G.; YAMANAKA, K. **In vitro fertilization genetic algorithm**. In: *Evolutionary Algorithms*, chapter 4, p. 57–68. InTech, 2011.
- [9] CHEN, Y.; GUAN, Z.; SHAO, X. **A comparative analysis of job scheduling algorithm**. In: *2011 International Conference on Management Science and Industrial Engineering*, p. 1091–1095. IEEE, 2011.
- [10] ENGELBRECHT, A. P. **Computational Intelligence: An Introduction**. Wiley, 2007, England.

- [11] FERROLHO, A.; CRISÓSTOMO, M.; WÓJCIK, R. **Job shop scheduling problems with genetic algorithms.** In: *International Conference on Computer Engineering and Systems*, p. 76–80. IEEE, 2007.
- [12] FRENCH, S. **Sequencing and scheduling: An Introduction to the Mathematics of the Job-Shop.** Ellis Horwood, West Sussex, England, 1982.
- [13] GAO, L.; ZHANG, G.; ZHANG, L.; LI, X. **An efficient memetic algorithm for solving the job shop scheduling problem.** *Computers and Industrial Engineering*, 60:699–705, 2011.
- [14] Gendreau, M.; Potvin, J.-Y., editors. **Handbook of Metaheuristics.** Springer, New York, 2010.
- [15] GLOVER, F. **Future paths for integer programming and links to artificial intelligence.** *Computers Operations Research*, 13:533–549, 1986.
- [16] GONÇALVES, J. F.; DE MAGALHÃES MENDES, J. J.; RESENDE, M. G. C. **A hybrid genetic algorithm for the job shop scheduling problem.** *European Journal of Operational Research*, 167:2005, 2002.
- [17] HOLLAND, J. H. **Adaptation in Natural and Artificial Systems.** MIT Press, 1992.
- [18] HUANG, M.; LIU, P.; LIANG, X. **An improved multi-population genetic algorithm for job shop scheduling problem.** In: *2010 IEEE International Conference on Progress in Informatics and Computing*, p. 272–275. IEEE, 2010.
- [19] JINGHUA, W.; MIANZHOU, C. **Research of an improved genetic algorithm for job shop scheduling.** In: *2010 International Conference on Measuring Technology and Mechatronics Automation*, p. 1076–1078. IEEE, 2010.
- [20] JONG, K. A. D. **Evolutionary computation: a unified approach.** MIT Press, Cambridge, 2006.
- [21] KAMMER, M.; VAN DEN AKKER, M.; HOOGEVEEN, H. **Identifying and exploiting commonalities for the job-shop scheduling problem.** *Computers and Operations Research*, 38:1556–1561, 2011.
- [22] LIN, T.-L.; HORNG, S.-J.; KAO, T.-W.; CHEN, Y.-H.; RUN, R.-S.; CHEN, R.-J.; LAI, J.-L.; KUO, I.-H. **An efficient job-shop scheduling algorithm based on particle swarm optimization.** *Expert Systems with Applications*, 37:2629–2636, 2010.
- [23] LIOUANE, N.; SAAD, I.; HAMMADI, S.; BONE, P. **Ant systems and local search optimization for flexible job shop scheduling production.** *International Journal of Computers, Communications and Control*, 2:174–184, 2007.

- [24] LIU, M.; JIANG SUN, Z.; WEI YAN, J.; SONG KANG, J. **An adaptive annealing genetic algorithm for the job-shop planning and scheduling problem.** *Expert Systems with Applications*, 38:9248–9255, 2011.
- [25] MA, J.; MA, J.; ZHU, Y.; SHI, G. **Solving the flexible job-shop scheduling problem by immune genetic algorithm.** In: *3rd International Conference on Advanced Computer Theory and Engineering*, volume 4, p. 215–218. IEEE, 2010.
- [26] MANIKAS, A.; CHANG, Y.-L. **Multi-criteria sequence-dependent job shop scheduling using genetic algorithms.** *Computers and Industrial Engineering*, 56:179–185, 2009.
- [27] MILLER, B. L.; GOLDBERG, D. E. **Genetic algorithms, tournament selection, and the effects of noise.** *Complex Systems*, 9:193–212, 1995.
- [28] NAZARATHY, Y.; WEISS, G. **A fluid approach to large volume job shop scheduling.** *Journal of Scheduling*, 13:509–529, 2010.
- [29] PARDALOS, P. M.; SHYLO, O. V. **An algorithm for the job shop scheduling problem based on global equilibrium search techniques.** *Computational Management Science*, 3:331–348, 2006.
- [30] PINEDO, M. L. **Scheduling: Theory, Algorithms, and Systems.** Springer, New York, 2008.
- [31] PING WANG, H.; SHI, L. **Improved genetic algorithm for solving the fuzzy multiobjective job shop problem.** In: *2010 IEEE 17Th International Conference on Industrial Engineering and Engineering Management*, p. 1542–1545. IEEE, 2010.
- [32] REGO, C.; DUARTE, R. **A filter-and-fan approach to the job shop scheduling problem.** *European Journal of Operational Research*, 194:650–662, 2009.
- [33] RONDON, R. L. A.; CARVALHO, A. S. **Solving a real job shop scheduling problem.** In: *35th Annual Conference of the IEEE Industrial Electronics Society*, p. 2494–2498. IEEE, 2009.
- [34] ROSHANA EI, V.; NADERI, B.; JOLAI, F.; KHALILI, M. **A variable neighborhood search for job shop scheduling with set-up times to minimize makespan.** *Future Generation Computer Systems*, 25:654–661, 2009.
- [35] SAIDI-MEHRABAD, M.; FATTAHI, P. **Flexible job shop scheduling with tabu search algorithms.** *The International Journal of Advanced Manufacturing Technology*, 32:563–570, 2007.

- [36] SNYDER, L. V.; DASKIN, M. S. **A random-key genetic algorithm for the generalized traveling salesman problem.** *European Journal of Operational Research*, 174:38–53, 2006.
- [37] TAILLARD, E. **Benchmarks for basic scheduling problems.** *European Journal of Operational Research*, 64:278–285, 1993.
- [38] TALBI, E.-G. **Metaheuristics: from design to implementation.** Wiley, Hoboken, New Jersey, 2009.
- [39] WHITLEY, D. **An overview of evolutionary algorithms: Practical issues and common pitfalls.** *Information and Software Technology*, 43:817–831, 2001.
- [40] WHITLEY, D. **Genetic algorithms and evolutionary computing.** Technical report, Computer Science Department, Colorado State University, 2002.
- [41] XIAOMEI, Y.; JIANCHAO, Z.; JIYE, L.; JIAHUA, L. **A genetic algorithm for job shop scheduling problem using co-evolution and competition mechanism.** In: *2010 International Conference on Artificial Intelligence and Computational Intelligence*, p. 133–136. IEEE, 2010.
- [42] XING, L.-N.; CHEN, Y.-W.; WANG, P.; ZHAO, Q.-S.; XIONG, J. **A knowledge-based ant colony optimization for flexible job shop scheduling problems.** *Applied Soft Computing*, 10:888–896, 2010.
- [43] YIN, H.; WANG, Y.; XIAO, N.; HU, E.; JIANG, Y. **A three-dimensional encoding genetic algorithm for job shop scheduling.** In: *2007 International Conference on Computational Intelligence and Security Workshops*, p. 175–178. IEEE, 2007.
- [44] ZHANG, G. **Using matrix-coded genetic algorithm for solving the flexible job-shop scheduling.** In: *2010 International Conference on Computational Intelligence and Software Engineering*, p. 1–4. IEEE, 2010.
- [45] ZHANG, G.; GAO, L.; SHI, Y. **A genetic algorithm and tabu search for multi objective flexible job shop scheduling problems.** In: *2010 International Conference on Computing, Control and Industrial Engineering*, p. 251–254. IEEE, 2010.
- [46] ZHANG, G.; SHAO, X.; LI, P.; GAO, L. **An effective hybrid particle swarm optimization algorithm for multi-objective flexible job-shop scheduling problem.** *Computers and Industrial Engineering*, 56:1309–1318, 2009.