

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

Éwerton Carlos de Araújo Assis

Dezembro 2011

1 Introdução

- Objetivos

2 Problemas de escalonamento job-shop

- Critérios de otimização
- Pressupostos em problemas de escalonamento job-shop
- Extensões à definição tradicional

3 Soluções heurísticas e metaheurísticas

- Padrões identificados nos métodos de solução metaheurísticos
 - Algoritmo auxiliar paralelo baseado na fertilização in vitro
 - População inicial

- 4 Solução metaheurística evolucionária
 - Algoritmos genéticos
 - Estratégias evolutivas
 - Hibridização em algoritmos evolucionários
- 5 Algoritmo evolucionário híbrido baseado na fertilização in vitro
 - Representação dos indivíduos
 - Mecanismos de seleção
 - Mecanismos de recombinação
 - Mecanismos de mutação
- 6 Análise dos resultados obtidos
 - Efetividade da solução
 - Influência dos mecanismos de seleção e de variabilidade

7 Conclusão

8 Referências Bibliográficas

Heurísticas e metaheurísticas como meios de resolver problemas de escalonamento job-shop

- Complexidade dos problemas de escalonamento job-shop e demais problemas de otimização

Heurísticas e metaheurísticas como meios de resolver problemas de escalonamento job-shop

- Complexidade dos problemas de escalonamento job-shop e demais problemas de otimização
- Importância das heurísticas e metaheurísticas em problemas de otimização

Heurísticas e metaheurísticas como meios de resolver problemas de escalonamento job-shop

- Complexidade dos problemas de escalonamento job-shop e demais problemas de otimização
- Importância das heurísticas e metaheurísticas em problemas de otimização
- Necessidade de hibridização das metaheurísticas

Heurísticas e metaheurísticas como meios de resolver problemas de escalonamento job-shop

- Complexidade dos problemas de escalonamento job-shop e demais problemas de otimização
- Importância das heurísticas e metaheurísticas em problemas de otimização
- Necessidade de hibridização das metaheurísticas
- Estudos de caso: efetividade e eficiência de uma solução metaheurística

Objetivos do presente trabalho

- Identificar problemas de escalonamento job-shop (PEJS) e suas definições

Objetivos do presente trabalho

- Identificar problemas de escalonamento job-shop (PEJS) e suas definições
- Identificar heurísticas e metaheurísticas utilizadas em PEJS

Objetivos do presente trabalho

- Identificar problemas de escalonamento job-shop (PEJS) e suas definições
- Identificar heurísticas e metaheurísticas utilizadas em PEJS
- Definir uma solução metaheurística híbrida baseada na fertilização in vitro (FIV-AG)

Objetivos do presente trabalho

- Identificar problemas de escalonamento job-shop (PEJS) e suas definições
- Identificar heurísticas e metaheurísticas utilizadas em PEJS
- Definir uma solução metaheurística híbrida baseada na fertilização in vitro (FIV-AG)
- Analisar a solução algorítmica construída e sua efetividade

Problemas de escalonamento job-shop (PEJS)

Definição formal de um problema de escalonamento job-shop:

- 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

Problemas de escalonamento job-shop (PEJS)

Definição formal de um problema de escalonamento job-shop:

- 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}

Problemas de escalonamento job-shop (PEJS)

Definição formal de um problema de escalonamento job-shop:

- 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}
- Cada tarefa i (J_i) apresenta um tempo de processamento para cada máquina, i.e. para cada operação o_{ij} há um tempo de processamento p_{ij}

Problemas de escalonamento job-shop (PEJS)

- Restrições tecnológicas determinam a ordem de processamento de cada tarefa através das m máquinas

Problemas de escalonamento job-shop (PEJS)

- 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

Problemas de escalonamento job-shop (PEJS)

Considerações relativas à definição:

- Necessidade de construir um escalonador de tarefas de forma a agendar as n tarefas através das m máquinas a partir de critérios de otimização estabelecidos

Problemas de escalonamento job-shop (PEJS)

Considerações relativas à definição:

- Necessidade de construir um escalonador de tarefas de forma a agendar as n tarefas através das m máquinas a partir de critérios de otimização estabelecidos
- O problema de escalonamento job-shop é considerado um problema numericamente intratável, NP-difícil (para $m \geq 2$) [2] e que apresenta um limite superior de $(n!)^m$ soluções possíveis

Critérios de otimização

- O critério básico tem como objetivo minimizar o C_{max} (*makespan*), i.e. minimizar o tempo total da tarefa de maior tempo de processamento

C_i é o tempo total de processamento da tarefa J_i , ou

$$C_i = r_i + \sum_{k=1}^m (W_{ik} + p_{ij(k)})$$

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

W_{ik} é o tempo de espera da tarefa J_i no processamento da operação k

Critérios de otimização

- A minimização do valor F_{max} tem por objetivo minimizar o tempo tempo gasto pelas tarefas na oficina

F_{max} o tempo total de processamento da tarefa que permaneceu por mais tempo na oficina

F_i é o tempo de processamento da tarefa i a partir do instante que esta encontra-se na oficina ($F_i = C_i - r_i$)

Critérios baseados na data de entrega

- Minimização do valor T_{max} é apropriada em contextos nos quais o atraso no processamento de tarefas apresenta alguma penalidade

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

T_i $T_i = \max\{C_i - d_i, 0\}$, o atraso da tarefa J_i ; e

E_i $E_i = \max\{d_i - C_i, 0\}$, a atencipação da tarefa J_i .

Critérios baseados na data de entrega

- Minimização do valor T_{max} é apropriada em contextos nos quais o atraso no processamento de tarefas apresenta alguma penalidade
- Minimização do valor E_{max} é apropriada nos contextos em que há um benefício ao antecipar-se o processamento das tarefas

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

T_i $T_i = \max\{C_i - d_i, 0\}$, o atraso da tarefa J_i ; e

E_i $E_i = \max\{d_i - C_i, 0\}$, a antecipação da tarefa J_i .

Critérios baseados na data de entrega

- Seja n_T o número de tarefas atrasadas, a minimização do valor n_T tem o propósito de diminuir o número de tarefas atrasadas

Critérios baseados em custos

- $N_w(t)$ 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

Critérios baseados em custos

- 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

Critérios baseados em custos

- 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
- Minimizar o número médio de tarefas completadas (\bar{N}_c), de forma a reduzir os custos dos bens produzidos

Critérios baseados em custos

- 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
- Minimizar o número médio de tarefas completadas (\bar{N}_c), de forma a reduzir os custos dos bens produzidos
- Maximizar o número médio de tarefas sendo processadas (\bar{N}_p) com o intuito de realizar um uso eficiente das máquinas

Pressupostos em problemas de escalonamento job-shop

- Cada tarefa é uma entidade

Pressupostos em problemas de escalonamento job-shop

- Cada tarefa é uma entidade
- Não existe preempção

Pressupostos em problemas de escalonamento job-shop

- Cada tarefa é uma entidade
- Não existe preempção
- Cada tarefa é constituída de m operações distintas, uma para cada máquina

Pressupostos em problemas de escalonamento job-shop

- Cada tarefa é uma entidade
- Não existe preempção
- Cada tarefa é constituída de m operações distintas, uma para cada máquina
- Tarefas não são canceladas

Pressupostos em problemas de escalonamento job-shop

- O tempo de processamento é independente da agenda construída

Pressupostos em problemas de escalonamento job-shop

- O tempo de processamento é independente da agenda construída
- Tarefas podem ter que esperar para serem processadas

Pressupostos em problemas de escalonamento job-shop

- O tempo de processamento é independente da agenda construída
- Tarefas podem ter que esperar para serem processadas
- Existe apenas um único tipo de cada máquina

Pressupostos em problemas de escalonamento job-shop

- O tempo de processamento é independente da agenda construída
- Tarefas podem ter que esperar para serem processadas
- Existe apenas um único tipo de cada máquina
- Máquinas podem estar ociosas

Pressupostos em problemas de escalonamento job-shop

- As máquinas não podem processar qualquer tarefa mais de uma vez

Pressupostos em problemas de escalonamento job-shop

- As máquinas não podem processar qualquer tarefa mais de uma vez
- As máquinas da oficina estão sempre disponíveis para processamento das tarefas

Pressupostos em problemas de escalonamento job-shop

- 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

Pressupostos em problemas de escalonamento job-shop

- 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
- Não existe aleatoriedade

Extensões à definição tradicional

PEJS flexível além de determinar a ordem de execução das tarefas, o escalonador deve determinar em qual das máquinas determinada tarefa será executada em dado momento

PEJS multi-objetivo estabelecem mais de um critério de otimização no modelo do problema a ser solucionado

Soluções heurísticas e metaheurísticas

Heurística “a arte de descobrir novas estratégias (...) para solucionar problemas” (tradução livre) [5];

Metaheurísticas “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) [3].

Soluções heurísticas e metaheurísticas

Metaheurísticas geralmente utilizadas como meio de resolver problemas de escalonamento job-shop:

- algoritmos genéticos;

Soluções heurísticas e metaheurísticas

Metaheurísticas geralmente utilizadas como meio de resolver problemas de escalonamento job-shop:

- algoritmos genéticos;
- algoritmos meméticos;

Soluções heurísticas e metaheurísticas

Metaheurísticas geralmente utilizadas como meio de resolver problemas de escalonamento job-shop:

- algoritmos genéticos;
- algoritmos meméticos;
- *particle swarm optimization*;

Soluções heurísticas e metaheurísticas

Metaheurísticas geralmente utilizadas como meio de resolver problemas de escalonamento job-shop:

- algoritmos genéticos;
- algoritmos meméticos;
- *particle swarm optimization*;
- *simulated annealing*;

Soluções heurísticas e metaheurísticas

Metaheurísticas geralmente utilizadas como meio de resolver problemas de escalonamento job-shop:

- algoritmos genéticos;
- algoritmos meméticos;
- *particle swarm optimization*;
- *simulated annealing*;
- *ant colony optimization*;

Soluções heurísticas e metaheurísticas

Metaheurísticas geralmente utilizadas como meio de resolver problemas de escalonamento job-shop:

- algoritmos genéticos;
- algoritmos meméticos;
- *particle swarm optimization*;
- *simulated annealing*;
- *ant colony optimization*;
- *variable neighborhood search*; e

Soluções heurísticas e metaheurísticas

Metaheurísticas geralmente utilizadas como meio de resolver problemas de escalonamento job-shop:

- algoritmos genéticos;
- algoritmos meméticos;
- *particle swarm optimization*;
- *simulated annealing*;
- *ant colony optimization*;
- *variable neighborhood search*; e
- busca tabu.

Padrões identificados nos métodos de solução metaheurísticos

- Métodos metaheurísticos híbridos

Padrões identificados nos métodos de solução metaheurísticos

- Métodos metaheurísticos híbridos
- As 82 instâncias disponíveis na OR-Library [1] e as 80 instâncias de Taillard [4] são frequentemente utilizadas como meios para estabelecer-se a qualidade da solução proposta frente aos resultados obtidos por outras soluções

Padrões identificados nos métodos de solução metaheurísticos

- Métodos metaheurísticos híbridos
- As 82 instâncias disponíveis na OR-Library [1] e as 80 instâncias de Taillard [4] são frequentemente utilizadas como meios para estabelecer-se a qualidade da solução proposta frente aos resultados obtidos por outras soluções
- Representação da solução através de *random-keys*, ou chaves aleatórias. Representação centrada na ordem de processamento das operações

Padrões identificados nos métodos de solução metaheurísticos

- Uso de operações de permutação como operador de busca local ou como operador de mutação em algoritmos evolucionários. Operadores de permutação:

Padrões identificados nos métodos de solução metaheurísticos

- Uso de operações de permutação como operador de busca local ou como operador de mutação em algoritmos evolucionários. Operadores de permutação:
 - inserção;

Padrões identificados nos métodos de solução metaheurísticos

- Uso de operações de permutação como operador de busca local ou como operador de mutação em algoritmos evolucionários. Operadores de permutação:
 - inserção;
 - inversão (geralmente de elementos adjacentes);

Padrões identificados nos métodos de solução metaheurísticos

- Uso de operações de permutação como operador de busca local ou como operador de mutação em algoritmos evolucionários. Operadores de permutação:
 - inserção;
 - inversão (geralmente de elementos adjacentes);
 - permutação ou troca de elementos em posições distintas; e

Padrões identificados nos métodos de solução metaheurísticos

- Uso de operações de permutação como operador de busca local ou como operador de mutação em algoritmos evolucionários. Operadores de permutação:
 - inserção;
 - inversão (geralmente de elementos adjacentes);
 - permutação ou troca de elementos em posições distintas; e
 - movimento de longa distância (mover uma sequência de elementos a uma determinada posição ou direção).

Solução metaheurística evolucionária

Componentes de uma solução algorítmica evolucionária:

- Representação da solução do problema como um indivíduo (solução–indivíduo)

Solução metaheurística evolucionária

Componentes de uma solução algorítmica evolucionária:

- Representação da solução do problema como um indivíduo (solução–indivíduo)
- População de indivíduos–soluções e desenvolvimento geracional dessa população

Solução metaheurística evolucionária

Componentes de uma solução algorítmica evolucionária:

- Representação da solução do problema como um indivíduo (solução–indivíduo)
- População de indivíduos–soluções e desenvolvimento geracional dessa população
- Conceitos biológicos como evolução, aptidão, recombinação e mutação genética influenciam essa solução algorítmica

Algoritmos genéticos

Os algoritmos genéticos em sua versão canônica desenvolvem a seguinte trajetória evolutiva sobre uma população:

- 1 são selecionados indivíduos—soluções da população corrente para gerar novos descendentes;

Algoritmos genéticos

Os algoritmos genéticos em sua versão canônica desenvolvem a seguinte trajetória evolutiva sobre uma população:

- 1 são selecionados indivíduos—soluções da população corrente para gerar novos descendentes;
- 2 novos indivíduos são gerados a partir de um mecanismo de recombinação genética (*crossover*);

Algoritmos genéticos

Os algoritmos genéticos em sua versão canônica desenvolvem a seguinte trajetória evolutiva sobre uma população:

- 1 são selecionados indivíduos—soluções da população corrente 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; e

Algoritmos genéticos

Os algoritmos genéticos em sua versão canônica desenvolvem a seguinte trajetória evolutiva sobre uma população:

- 1 são selecionados indivíduos—soluções da população corrente 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; e
- 4 a população corrente é substituída pelos descendentes gerados, dando origem a uma nova geração.

Estratégias evolutivas

Características próprias às estratégias evolutivas:

- 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

Estratégias evolutivas

Características próprias às estratégias evolutivas:

- 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
- há um foco importante no controle do *step-size*, ou à maneira que o algoritmo percorre o espaço de busca — controle feito a partir do operador de variabilidade (mutação)

Hibridização em algoritmos evolucionários

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.

- Necessidade de superar deficiências nos algoritmos evolucionários

Hibridização em algoritmos evolucionários

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.

- Necessidade de superar deficiências nos algoritmos evolucionários
- Explorar novas possibilidades de utilização; ampliando a efetividade e eficiência da solução algorítmica

Hibridização em algoritmos evolucionários

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.

- Necessidade de superar deficiências nos algoritmos evolucionários
- Explorar novas possibilidades de utilização; ampliando a efetividade e eficiência da solução algorítmica
- Propor novas soluções metaheurísticas a partir de conceitos já preestabelecidos

Algoritmo auxiliar paralelo baseado na fertilização in vitro

- Os melhores indivíduos-soluções de uma população tem em sua estrutura genética/cromossômica características que os tornam indivíduos de notável qualidade

Algoritmo auxiliar paralelo baseado na fertilização in vitro

- Os melhores indivíduos-soluções de uma população tem em sua estrutura genética/cromossômica características que os tornam indivíduos de notável qualidade
- Fluxograma conceitualmente simples:

Algoritmo auxiliar paralelo baseado na fertilização in vitro

- Os melhores indivíduos-soluções de uma população tem em sua estrutura genética/cromossômica características que os tornam indivíduos de notável qualidade
- Fluxograma conceitualmente simples:
 - coleta;

Algoritmo auxiliar paralelo baseado na fertilização in vitro

- Os melhores indivíduos-soluções de uma população tem em sua estrutura genética/cromossômica características que os tornam indivíduos de notável qualidade
- Fluxograma conceitualmente simples:
 - coleta;
 - manipulação genética; e

Algoritmo auxiliar paralelo baseado na fertilização in vitro

- Os melhores indivíduos-soluções de uma população tem em sua estrutura genética/cromossômica características que os tornam indivíduos de notável qualidade
- Fluxograma conceitualmente simples:
 - coleta;
 - manipulação genética; e
 - transferência.

Algoritmo auxiliar paralelo baseado na fertilização in vitro

- Os melhores indivíduos-soluções de uma população tem em sua estrutura genética/cromossômica características que os tornam indivíduos de notável qualidade
- Fluxograma conceitualmente simples:
 - coleta;
 - manipulação genética; e
 - transferência.
- Propósito principal: controlar a convergência da solução algorítmica sem denegrir os resultados obtidos

Algoritmo evolucionário híbrido baseado na fertilização in vitro

- Duas soluções algorítmicas foram construídas:

Algoritmo evolucionário híbrido baseado na fertilização in vitro

- Duas soluções algorítmicas foram construídas:
 - Algoritmo evolucionário híbrido baseado na fertilização in vitro (IVF/EV-JB); e

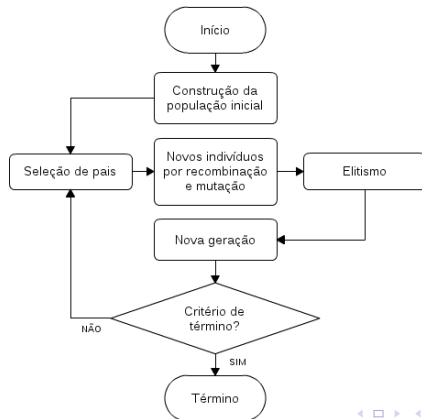
Algoritmo evolucionário híbrido baseado na fertilização in vitro

- Duas soluções algorítmicas foram construídas:
 - Algoritmo evolucionário híbrido baseado na fertilização in vitro (IVF/EV-JB); e
 - Algoritmo genético–evolucionário com conceitos canônicos (GA/EV-JB).

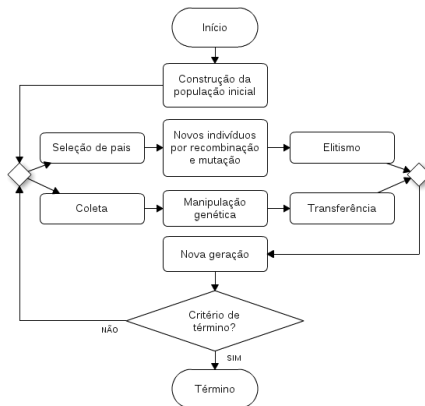
Algoritmo evolucionário híbrido baseado na fertilização in vitro

- Duas soluções algorítmicas foram construídas:
 - Algoritmo evolucionário híbrido baseado na fertilização in vitro (IVF/EV-JB); e
 - Algoritmo genético-evolucionário com conceitos canônicos (GA/EV-JB).
- Cada um dos algoritmos foram utilizados sob configurações diferentes a fim de obter-se um comparativo das soluções construídas

Algoritmo genético–evolucionário com conceitos canônicos (GA/EV-JB)



Algoritmo evolucionário híbrido baseado na fertilização in vitro (IVF/EV-JB)



Representação dos indivíduos

Representação por chaves aleatórias (*random-keys*):

Seja $n = 2$ e $m = 4$, $S = \langle 0.2, 0.5, 1.8, 6.7, 3.3, 2.4, 3.5, 2.4 \rangle$

Representação dos indivíduos

Representação por chaves aleatórias (*random-keys*):

Seja $n = 2$ e $m = 4$, $S = \langle 0.2, 0.5, 1.8, 6.7, 3.3, 2.4, 3.5, 2.4 \rangle$

① $\langle 0.2, 0.5, 1.8, 6.7, 3.3, 2.4, 3.5, 2.4 \rangle \rightarrow \langle 1, 2, 3, 8, 6, 4, 7, 5 \rangle;$

Representação dos indivíduos

Representação por chaves aleatórias (*random-keys*):

Seja $n = 2$ e $m = 4$, $S = \langle 0.2, 0.5, 1.8, 6.7, 3.3, 2.4, 3.5, 2.4 \rangle$

① $\langle 0.2, 0.5, 1.8, 6.7, 3.3, 2.4, 3.5, 2.4 \rangle \rightarrow \langle 1, 2, 3, 8, 6, 4, 7, 5 \rangle;$

② $\langle 1, 2, 3, 8, 6, 4, 7, 5 \rangle \rightarrow \langle 2, 1, 2, 1, 1, 1, 2, 2 \rangle,$

com $S'_i = (S_i \bmod n) + 1;$

Representação dos indivíduos

Representação por chaves aleatórias (*random-keys*):

Seja $n = 2$ e $m = 4$, $S = \langle 0.2, 0.5, 1.8, 6.7, 3.3, 2.4, 3.5, 2.4 \rangle$

- ① $\langle 0.2, 0.5, 1.8, 6.7, 3.3, 2.4, 3.5, 2.4 \rangle \rightarrow \langle 1, 2, 3, 8, 6, 4, 7, 5 \rangle$;
- ② $\langle 1, 2, 3, 8, 6, 4, 7, 5 \rangle \rightarrow \langle 2, 1, 2, 1, 1, 1, 2, 2 \rangle$,
com $S'_i = (S_i \bmod n) + 1$;
- ③ $S_f = \langle o_{2,1}, o_{1,1}, o_{2,2}, o_{1,2}, o_{1,3}, o_{1,4}, o_{2,3}, o_{2,4} \rangle$;
 $o_{i,k}$: operação da tarefa i na k -ésima máquina de sua configuração tecnológica.

População inicial

- Os indivíduos-soluções da população inicial são criados a partir de distribuições normais

População inicial

- 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 $N(0, n \times m)$; n o número de tarefas e m o número de máquinas da oficina

População inicial

- 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 $N(0, n \times m)$; n o número de tarefas e m o número de máquinas da oficina
- Tamanho da população (μ): $\mu = 2 \times n \times m$.

Mecanismos de seleção

Mecanismos de seleção utilizados:

- Seleção por ranqueamento linear;

Mecanismos de seleção

Mecanismos de seleção utilizados:

- Seleção por ranqueamento linear;
- Seleção proporcional à aptidão; e

Mecanismos de seleção

Mecanismos de seleção utilizados:

- Seleção por ranqueamento linear;
- Seleção proporcional à aptidão; e
- Seleção por torneio.

Mecanismos de recombinação

Mecanismos de recombinação utilizados:

- Recombinação de 1-ponto;

Mecanismos de recombinação

Mecanismos de recombinação utilizados:

- Recombinação de 1-ponto;
- Recombinação de n -pontos; e

Mecanismos de recombinação

Mecanismos de recombinação utilizados:

- Recombinação de 1-ponto;
- Recombinação de n -pontos; e
- Recombinação uniforme.

Mecanismos de seleção

Dois mecanismos de mutação foram utilizados:

- Mutação por permutação; e

Mecanismos de seleção

Dois mecanismos de mutação foram utilizados:

- Mutação por permutação; e
- Geração aleatória de indivíduos

Mecanismos de seleção

Dois mecanismos de mutação foram utilizados:

- Mutação por permutação; e
- Geração aleatória de indivíduos
 - Os 20% piores indivíduos da população são substituídos por novos indivíduos construídos aleatoriamente

Análise dos resultados obtidos

- Foram construídos 27 casos de experimentos a partir dos operadores de seleção e de variabilidade selecionados

Análise dos resultados obtidos

- Foram construídos 27 casos de experimentos a partir dos operadores de seleção e de variabilidade selecionados
- As soluções algorítmicas têm como critério de parada a criação de 500 gerações populacionais, com exceção da população inicial

Efetividade da solução

- Ambas as soluções desenvolvidas são soluções de qualidade por apresentarem respostas próximas ou semelhantes àsquelas encontradas pela literatura

Efetividade da solução

- Ambas as soluções desenvolvidas são soluções de qualidade por apresentarem respostas próximas ou semelhantes àsquelas encontradas pela literatura
- A média populacional e os valores de pior indivíduo do IVF/EV-JB corroboram com a motivação do algoritmo auxiliar paralelo (AAP) em manter qualidades genotípicas na população e prover resultados de qualidade

Influência dos mecanismos de seleção e de variabilidade

- Não foi possível identificar quais são as reais influências dos operadores de variabilidade e de seleção sobre a efetividade da solução

Influência dos mecanismos de seleção e de variabilidade

- Não foi possível identificar quais são as reais influências dos operadores de variabilidade e de seleção sobre a efetividade da solução
- Os resultados obtidos corroboram com o que é defendido na literatura: não existem operadores de seleção e variabilidade ótimos *per se*

Influência dos mecanismos de seleção e de variabilidade

- Não foi possível identificar quais são as reais influências dos operadores de variabilidade e de seleção sobre a efetividade da solução
- Os resultados obtidos corroboram com o que é defendido na literatura: não existem operadores de seleção e variabilidade ótimos *per se*
- As melhores configurações para a solução algorítmica construída são aquelas apresentadas nos experimentos 1, 2, 3, 25, 26, e 27

Conclusão

- Design de uma solução evolucionária
- Importância da hibridização em contextos específicos
- Importância da representação da solução
- O algoritmo auxiliar paralelo baseado na fertilização in vitro e suas características identificadas

Referências Bibliográficas I



BEASLEY, J. E.

Or-library.

<http://people.brunel.ac.uk/~mastjjb/jeb/info.html>, Dezembro 2011.



FRENCH, S.

Sequencing and scheduling: An Introduction to the Mathematics of the Job-Shop.

Ellis Horwood, West Sussex, England, 1982.



Gendreau, M.; Potvin, J.-Y., editors.

Handbook of Metaheuristics.

Springer, New York, 2010.

Referências Bibliográficas II



TAILLARD, E.

Benchmarks for basic scheduling problems.

European Journal of Operational Research, 64:278–285, 1993.



TALBI, E.-G.

Metaheuristics: from design to implementation.

Wiley, Hoboken, New Jersey, 2009.