Meta-heurística híbrida com *multi-start* aplicada no problema de balanceamento de linha de produção

Mateus Matos Rizzi
Universidade Federal do ABC – UFABC
Av. dos Estados, 5001, Bangú, CEP 09210-580, Santo André, SP, Brasil
m.rizzi@ufabc.edu.br

Antonio Augusto Chaves
Universidade Federal de São Paulo – UNIFESP
Rua Talin, 330, CEP 12231-280, São José dos Campos, SP, Brasil
antonio.chaves@unifesp.br

RESUMO

Esse trabalho visa verificar a performance do método *multi-start* quando utilizado em conjunto com uma meta-heurística híbrida que utiliza Algoritmo Genético de Chaves Aleatórias Viciadas (BRKGA) e método de agrupamento (CS) . Para validar, foi aplicado no problema de balanceamento de linha de produção (SALB-1) e comparado com os resultados obtidos pelo algoritmo utilizando apenas o BRKGA. As soluções são representadas como chaves aleatórias e divididas em grupos por assimilaridade, e realizadas buscas locais nos grupos mais promissores. Os resultados mostram a superioridade do método *multi-start* para encontrar as melhores soluções.

PALAVRAS CHAVE. multi-start, BRKGA+CS, SALB-1, metaheurísticas.

Área Principal: Metaheurísticas

ABSTRACT

This work verify the results of hybrid metaheuristic with multi-start when it were applied on the problem Assembly Line Balancing, SALB-1, and to validate were compared with the results of algorithm that uses Biased Random Key Genetic Algorithm .The hybrid metaheuristic uses a combination of Biased Random Key Genetic Algorithm and Clustering Search. Solutions were created by random keys and sorted by groups according to its relatedness of your keys, and the better groups were made local search. The results show superiority of multi-start to find better solution.

KEYWORDS. multi-start, BRKGA+CS, SALB-1, metaheuristics.

Main Area: Metaheuristics

1. Introdução

Problemas com espaços de soluções grandes tendem a ter uma maior probabilidade de ficarem presos em ótimos locais. Utilizando o método *multi-start* é possível reiniciar o processo de busca, evitando que a busca fique presa em espaços soluções pouco efetivos. Para testar a forma como o método escapa de espaços de soluções não promissores, foi escolhido o problema de balanceamento de linha de produção (do inglês *Assembly Line balancing* ou SALB).

O SALB é um problema classificado como *NP-difícil*, consistindo em uma sequência m de estações de trabalhos que precisam realizar um numero n de tarefas que possuem relações de precedência e que necessitam de um tempo t_i para ser realizadas. As estações de trabalho têm uma jornada de trabalho, tempo de ciclo C. Portanto, as estações apenas suportam uma quantidade de tarefas que a soma dos tempos não seja maior que C. O SALB é composto por duas vertentes, o SALB-1 tem um tempo C fixo e seu objetivo é descobrir o menor número de estações possíveis para realizar todas as tarefas da linha de produção. O SALB-2 tem um número fixo de estações, e o objetivo é encontrar o menor tempo de ciclo C. Neste trabalho abordaremos o SALB-1

Na literatura encontra-se uma variedade de artigos disponíveis que utilizaram diferentes procedimentos heurísticos e exatos para a resolução do SALB-1. [Baybars, 1986] e [Ghosh e Gagnon, 1989] utilizaram o procedimento de *branch & bound*. [Bock e Rosenberg, 1998] realizaram uma implementação paralela utilizando o programa SALOME [Scholl e Scholl, 1999]. [Scholl e Voß, 1997] e [Gen et al., 1996] utilizaram heurísticas que incluem procedimento por prioridade como o *Tabu Search*. [Gonçalves e De Almeida, 2002] utiliza o procedimento por prioridade com a meta-heurística BRKGA (do inglês *Biased Random Key Genetic Algorithm*) e heurísticas de buscas locais. Este método obtém as soluções ótimas nos 64 casos testados.

Neste trabalho propomos combinar a meta-heurística BRKGA com o método de agrupamento CS. O CS procura combinar meta-heurísticas e heurísticas de busca local, de tal forma que a busca seja intensificada somente em regiões promissoras do espaço de soluções. O agrupamento realizado pelo CS utiliza a população criada pelo BRKGA e divide em c grupos de chaves aleatórias, e toda vez que uma solução nova é criada, é enviada para o CS, onde é calculada a distância euclidiana das chaves aleatórias em relação aos centros, determinando qual grupo a solução de chaves será assimilado. Quando um grupo supera um parâmetro λ de soluções atribuídas ao grupo, se torna promissor e é realizado uma busca local no seu centro. Mas Caso seja feito um número β de buscas locais em um agrupamento sem melhora, esse centro é descartado e criado uma nova solução.

Métodos que procuram encontrar soluções ótimas precisam de uma grande diversidade de soluções para escapar de áreas de ótimos locais [Gonçalves e De Almeida, 2002]. Inúmeras heurísticas já foram propostas para que se possa evitar ótimos locais e aumentar a diversidade de solução. A heurística *multi-start* propõe que para aumentar a diversidade de soluções, recomece do zero os métodos por um número ω de vezes depois que γ iterações foram realizadas sem melhora.

O objetivo deste trabalho é utilizar o BRKGA com procedimento por prioridade proposto por [Gonçalves e De Almeida, 2002] e combina-lo no método híbrido *BRKGA+CS* em conjunto com a estratégia *multi-start*. Para avaliar o comportamento do método proposto serão testadas as 64 instâncias disponíveis na literatura, analisando a diferença dos resultados obtidos quando se usa o método *BRKGA* sem *multi-start*.

O restante do artigo está organizado na seguinte forma. Na seção 2 será abordado as heurísticas e meta-heurísticas utilizadas nesse projeto. Na seção 3 será explicada a decodificação para a geração da solução do problema. Na seção 4 será explicado as buscas locais usadas. Na seção 5 mostraremos os resultados computacionais obtidos pelo algoritmo que utiliza *multi-start* com BRKGA+CS, comparando com os resultados da literatura. Na seção 6 são explanadas as conclusões finais.

2. Heurísticas e Meta-heurísticas

2.1. BRKGA

Algoritmo Genético de Chaves Aleatórias Viciadas (BRKGA, do inglês *Biased Random Key Genetic Algorithm*), foi proposto por [Gonçalves e De Almeida, 2002] e [Gonçalves e Resende, 2004], baseada nos algoritmos genéticos (GA, do inglês *Genetic Algorithm*) [Holland, 1975] que se utilizam das ideias da teoria da evolução proposta por Charles Darwin:

1. Avaliação: critério de verificação da aptidão dos indivíduos (soluções) da população;
2. Seleção: define-se um critério para selecionar indivíduos para reprodução, onde os mais adaptados (melhores soluções) têm uma probabilidade maior de se reproduzir do que soluções ruins;
3. Cruzamento: componente responsável por combinar características de diferentes soluções;
4. Mutação: permite adicionar variedade à população de soluções;
5. Atualização: os novos indivíduos são inseridos na população de soluções.

[Bean, 1994] propôs o Algoritmo Genético com Chaves Aleatórias (RKGA, do inglês *Random Key Genetic Algorithm*), que representa as soluções como vetores de chaves aleatórias de números reais no intervalo [0,1]. Com isso, os problemas se tornaram adaptados para serem solucionados a partir da interpretação de chaves aleatórias e sendo possível dar um valor de *fitness* para cada solução gerada, determinando quais tipos de chaves aleatórias geram boas soluções. No *BRKGA* a diferença em relação aos demais algoritmos genéticos é a criação das gerações, sendo dividido em três operações distintas:

1. Copiar a população elite p_e para a nova geração;
2. Realizar o cruzamento de alelos entre um indivíduo da p_e com um indivíduo não elite (p – p_e). Tal troca de alelo é realizada através de um sorteio, no qual o alelo do indivíduo da população elite tem uma chance $rhoe$ de ser escolhido. A combinação dos pais é realizada com o método de cruzamento uniforme parametrizado, onde o filho herda a i-ésima chave do pai elite com probabilidade $rhoe > 0,5$ e a do pai não-elite com probabilidade 1-$rhoe$.
3. Criar um número pm de mutantes, soluções com chaves aleatórias geradas aleatoriamente.

2.2. *Clustering Search*(CS)

[Oliveira et al., 2013] propuseram o método híbrido *Clustering Search* (CS), que verifica grupos de soluções que possuem maior incidência e realiza buscas locais nessas regiões promissoras. Para seu funcionamento, a meta-heurística realiza três etapas: gerador de soluções, processo de agrupamento e heurística de busca local.

No gerador de soluções, é utilizado heurísticas que quantifica soluções. No processo de agrupamento é criado soluções representativas para o espaço de busca (*clusters*), onde é verificado a maior região de incidência para se realizar uma heurística de busca Local. Selecionado o *cluster* que a solução mais se assemelha, utiliza-se o método *Path Relinking* (PR) Glover et al. [2000] ou um operador de cruzamento, para os grupos assimilarem as soluções geradas. Caso o *cluster* tenha um valor de solução melhor, a assimilaridade será baixa, caso contrário, se torna alta. Caso a região seja incidente ao chamar as buscas locais e não obtenha nenhuma melhora, depois de β vezes, o centro desse grupo é descartado e gerado um novo.

2.3. BRKGA+CS

O método híbrido combina a população de soluções por chaves aleatórias gerados pelo método BRKGA e divide em grupos, que são determinados através da semelhança de chaves aleatórias, para a melhor definição de locais promissores e realizar buscas locais apenas em áreas de solução que possuem chaves que apresentam soluções promissoras. O BRKGA funciona paralelamente e a cada geração é mandado os filhos gerados para o CS, conforme mostrado na Figura 1. O processo é divido em cinco etapas distintas:

1. A primeira etapa do *BRKGA+CS* é a criação dos centros dos *clusters* com soluções geradas aleatoriamente. As soluções são formadas por chaves aleatórias, permitindo o cálculo de distância euclidiana para calcular a similaridade e tornar os processos do CS, exceto a busca local, independente do problema abordado.

2. A segunda etapa consiste no processo de evolução do BRKGA, que servirá como gerador de soluções para o agrupamento do CS. A população inicial é enviada para o agrupamento e nas gerações seguintes a população elite é copiada e é incrementada com soluções filhas, através de processo de cruzamento parametrizados e criação de soluções novas. Apenas as soluções filhas são enviadas para o agrupamento nas gerações seguintes.

3. A terceira etapa consiste no processo de agrupamento de cada filho ao *cluster* mais similar de acordo com a menor distância calculada. Selecionado o grupo, a solução causa uma perturbação ao centro, atualizando o centro com a combinação entre suas chaves aleatórias. Para essa etapa utiliza-se métodos como o *Path-Relinking* (PR) para analisar o caminho entre o centro do *cluster* e a solução agrupada ou aplica-se o cruzamento uniforme parametrizado nas soluções. Com o PR ou o cruzamento parametrizado, o novo centro será a melhor solução obtida neste caminho, mesmo que esta seja pior que o centro atual, mas permitirá uma maior diversidade no espaço solução do problema.

4. Quando o número de soluções agrupadas em um *cluster* atinge o limitante λ, seu centro é decodificado em uma solução do problema e esta é enviada para o componente de busca local.

5. Após a busca local, é verificado se foi encontrado uma solução melhor, em caso de melhora o centro é atualizado. Mas se um número β de buscas locais são realizadas sem melhora do centro, esse centro é descartado e criado um novo.

A combinação das meta-heurísticas já foi abordada em outros problemas na literatura e tem apresentados resultados robustos como em [Pomari e Chaves, 2014] que aborda alocação de navios em berços e obteve todas as soluções ótimas para as instâncias testadas.

2.4. Multi-start

[Michel Gendreau, 2010] definem meta-heurísticas como sendo algoritmos que provem técnicas para integrar e coordenar heurísticas para resolver problemas de otimização. Esses métodos caracterizam-se em estratégias que procuram escapar de locais ótimos e tem uma robusta pesquisa de espaços promissores de soluções, procurando soluções melhores na vizinhança da solução pesquisada.

Multi-start explora a vizinhança recomeçando aleatoriamente a solução inicial, dando assim a diversidade para escapar de locais ótimos. Sem essa diversificação, os métodos heurísticos podem ser confinados a uma pequena área do espaço de solução, se tornando impossível encontrar a região da solução ótima global. Nesse artigo foi utilizado o *multi-start* em conjunto com a meta-heurística híbrida BRKGA+CS, reinicializando um número ω de vezes o funcionamento depois de um número γ de gerações sem melhora.

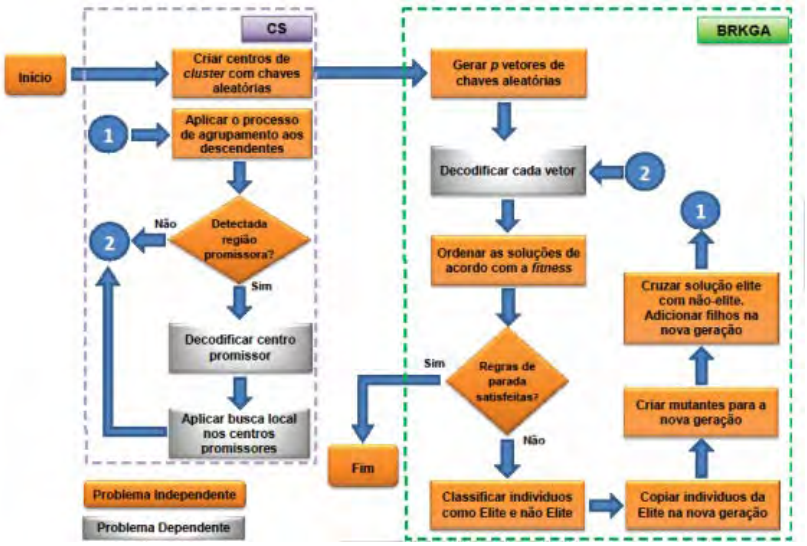

Figura 1: Fluxograma do Brkga+CS [Pomari e Chaves, 2014]

3. Decodificação do Salb-1 em chaves aleatórias

SALB-1 é um problema *NP-difícil*. Portanto, os procedimentos heurísticos precisam ser robustos para evitar que tenha pouca variedade de soluções viáveis. [Gonçalves e De Almeida, 2002] propõe uma heurística de prioridade por operação orientada que utiliza de quatro etapas para determinar a ordem de tarefas:

1. Faça uma lista de operações com todas as precedências concluídas.
2. Verifique qual dessas tarefas não excedem o tempo limite da estação escolhida. Caso nenhuma tarefa não exceda o limite de tempo, uma estação nova é criada.
3. Dentro das tarefas que não excedam o tempo da estação, escolha a que tem a maior chave aleatória.
4. Repita as etapas 1 à 3 até todas as operações sejam alocadas.

Heurísticas de prioridade são classificadas em duas vertentes, conforme explicitado em [Talbot et al., 1986]:

1. Estado orientado: Ordena a prioridade de todas as tarefas e aloca as tarefas conforme sua ordem de prioridade.
2. Operação orientada: Para cada iteração verifica todas as operações disponíveis e aquela que tiver maior prioridade é a escolhida

[Scholl et al., 1994] comparou inúmeras heurísticas de prioridade e concluiu que heurísticas por operação orientada são superiores as de estado orientado.

Em [Gonçalves e De Almeida, 2002] é dado um exemplo de como funciona a heurística de prioridade para o SALB-1, como mostrado na Figura 2. Um problema com 11 tarefas e estações com carga horária de 2,5. Para efeitos práticos, o tempo de processamento de tarefas também são as chaves aleatórias.

Na primeira iteração, a lista de operações disponíveis são A, C, D. É verificado que todas as tarefas não ultrapassem o tempo de processamento da estação, já que a estação está vazia. A operação D é escolhida por ter a chave maior, portanto, maior prioridade. As interações 2 a 4

fazem o mesmo procedimento. Na quinta iteração as operações disponíveis são B, K, mas apenas a operação K está apta por não exceder a carga horária da estação. Na sexta iteração, apenas a tarefa B está disponível e seu tempo de processamento excede a carga horária da estação (porque 2,4 + 0,6 excede 2,5), então uma nova estação é criada. A Tabela 1 apresenta todos os detalhes de cada iteração.

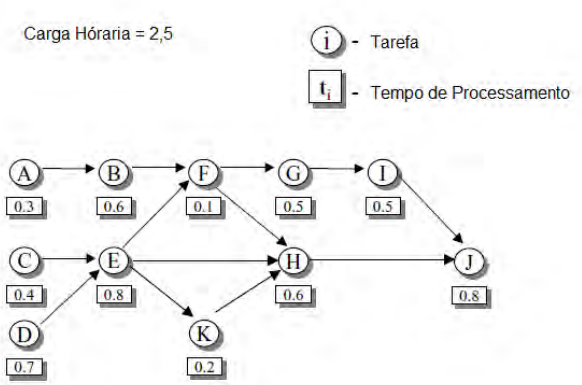

Figura 2: Exemplo do problema de linha de produção adaptado de [Gonçalves e De Almeida, 2002]

Tabela 1: Aplicação da heurística de prioridade por Operação orientada no problema de linha de produção

Iteração	Estação de trabalho	Tarefas disponíveis	Operação escolhida	Tempo Acumulado
1	1	A-0,3 C-0,4 D 0,7	D	0,7
2		A-0,3 C-0,4	C	1,1
3		E-0,8 A-0,3	E	1,9
4		A-0,3 K-0,2	A	2,2
5		B-0,6 K-0,2	K	2,4
6	2	B-0,6	B	0,6
7		F-0,1	F	0,7
8		G-0,5 H-0,6	H	1,3
9		G-0,5	G	1,8
10		I-0,5	I	2,3
11	3	J-0,8	J	0,8

4. Buscas Locais

Quando um centro que compõe o CS tem soluções semelhantes superior ao parâmetro λ, o centro se torna promissor, sendo realizado uma busca local nessa solução. A busca local usada nesse artigo é feita usando três heurísticas distintas, *Shifdown*, *Shiftup* e Troca:

Shifdown: Nessa heurística é feito a retirada da última tarefa de uma estação e colocada em outra estação que permita a inserção dessa tarefa, verificando precedência e tempo. Esse procedimento é realizado para todas as estações, seguindo a ordem da última estação criada até a primeira e só é finalizado a busca quando a estação não tem mais tarefa alocada ou nenhuma troca pode ser feita.

Shiftup: Essa heurística tem o mesmo funcionamento da heurística *Shifdown*, a diferença é que a análise de estação, é da primeira estação criada até a última estação criada

Troca: Realiza-se uma troca na posição de tarefas de estações diferentes, se não violar a precedência de tarefas, a troca é feita.

5. Resultados computacionais

O método heurístico foi codificado na linguagem C++ e os testes computacionais foram executados em um PC com processador Intel core i5, 2.5 GHz e memória de 6 GB. Os experimentos foram realizados para mostrar a eficácia e a robustez do método híbrido *BRKGA+CS* usando *multi-start*.

Os experimentos computacionais foram conduzidos com exemplares disponíveis em `http://alb.mansci.de/index`. Todos os problemas do *SALB-1* já tem a sua otimização alcançada, o que busca empreender nesse método é mostrar a robustez do método híbrido com *multi-start* em encontrar soluções robustas e otimizadas.

Os parâmetros do *CS+BRKGA multi-start* foram calibrados através de testes empíricos realizados sobre um subconjunto de instâncias. Os parâmetros adotados neste trabalho são:

- Número de *clusters* do *CS* (NC) = 20.
- Limitante de *cluster* promissor (λ) = 25.
- Intensidade da perturbação de um *cluster* (β) = 50.
- População(p) = 100.
- Fração de população elite(p_e) = 0,1.
- Fração de população mutante (p_m) = 0,2.
- Probabilidade de um alelo elite ser escolhido ($rhoe$) = 0,7.
- Número de gerações no *BRKGA* = 100.
- Número de recomeços (ω) = 5.
- Limitante de gerações sem melhora (γ) = 50.

Os resultados computacionais obtidos encontram-se na Tabela **??** . Estas tabelas contêm as seguintes colunas:

- Instância: nome da instância;
- Tarefas : números de tarefas de cada instância
- Tempo de Ciclo: Tempo que cada estação suporta
- Opt: melhor solução conhecida na literatura;
- Sol*: melhor solução encontrada;
- Sol: solução média em 20 execuções;
- DRP: desvio relativo percentual,dado por (Sol*-Opt)*100/Opt;
- T*: tempo médio para obter a melhor solução;
- T: tempo total de execução do método.

Na Tabela 2 cada linha apresenta o resultado de um dos 60 de exemplares utilizados nos testes computacionais. Para cada exemplar o método foi executado 20 vezes.

Tabela 2: SALB-1: Comparação de resultados

			BRKGA					BRKGA + CS Multi-start				
Nome	Tarefas	Tempo de ciclo	Opt	Sol*	SOI	DRP	T*	T	Sol*	SOI	DRP	T*	T
BOWMAN	5	20	5	5	5,40	0,00	0,00	0,36	5	5	0,00	2,10	2,62
HESKIA	28	205	5	6	6,25	20,00	0,00	3,43	6	6	20,00	24,99	31,19
	28	216	5	5	5,80	0,00	0,00	3,59	5	5	0,00	25,90	32,33
	28	256	4	5	5,00	25,00	0,00	3,62	5	5	25,00	26,89	33,55
	28	324	4	4	4,00	0,00	0,00	3,60	4	4	0,00	26,97	33,66
	28	342	3	4	4,00	33,33	0,00	3,59	4	4	33,33	26,97	33,66
	28	138	8	9	9,80	12,50	0,00	3,54	8	8	0,00	25,35	31,63
Arc	83	5048	16	16	16,90	0,00	0,00	48,50	16	16	0,00	379,19	473,35
	83	5853	14	14	14,90	0,00	0,00	50,68	14	14	0,00	392,56	490,08
	83	6842	12	12	12,75	0,00	0,00	50,91	12	12	0,00	398,04	496,89
	83	7571	11	11	11,00	0,00	0,00	50,83	11	11	0,00	408,23	509,53
	83	8414	10	10	10,00	0,00	0,00	50,84	10	10	0,00	410,77	512,81
	83	8998	9	9	9,10	0,00	0,00	50,85	9	9	0,00	407,09	508,17
	83	10816	8	8	8,00	0,00	0,00	50,81	8	8	0,00	409,79	511,50
JACKSON	11	13	4	4	4,60	0,00	0,00	3,04	4	4	0,00	23,60	29,46
	11	14	4	4	4,25	0,00	0,00	0,68	4	4	0,00	4,43	5,53
	11	21	3	3	3,00	0,00	0,00	0,57	3	3	0,00	3,54	4,41
	11	7	8	9	9,25	12,50	0,00	0,56	9	9	12,50	3,44	4,30
	11	9	6	7	7,50	16,67	0,00	0,55	7	7	16,67	3,38	4,21
	11	10	5	6	6,20	20,00	0,00	0,56	6	6	20,00	3,44	4,30
JAESCHKE	9	18	3	3	3,00	0,00	0,00	0,43	3	3	0,00	2,63	3,27
	9	6	8	8	8,15	0,00	0,00	0,43	8	8	0,00	2,57	3,21
	9	7	7	8	8,00	14,29	0,00	0,42	8	8	14,29	2,58	3,22
	9	8	6	7	7,00	16,67	0,00	0,42	7	7	16,67	2,59	3,22
	9	10	4	5	5,00	25,00	0,00	0,42	5	5	25,00	2,58	3,22
KILBRID	45	57	10	11	11,80	10,00	0,00	10,00	10	10,5	10,00	78,55	97,03
	45	79	7	8	8,55	14,29	0,00	10,49	8	8	14,29	78,81	98,32
	45	92	6	7	7,30	16,67	0,00	10,54	7	7	16,67	80,27	100,18
	45	110	6	6	6,00	0,00	0,00	10,56	6	6	0,00	81,85	102,15
	45	138	4	5	5,00	25,00	0,00	10,53	5	5	25,00	82,06	102,42
	45	184	3	4	4,00	33,33	0,00	10,52	4	4	33,33	82,72	103,30
MERTENS	7	18	2	2	2,00	0,00	0,00	0,87	2	2	0,00	6,19	7,72
	7	6	6	6	6,35	0,00	0,00	0,38	6	6	0,00	2,34	2,92
	7	7	5	6	6,00	20,00	0,00	0,36	6	6	20,00	2,17	2,70
	7	8	5	5	5,35	0,00	0,00	0,33	5	5	0,00	2,06	2,57
	7	10	3	4	4,05	33,33	0,00	0,37	4	4	33,33	2,15	2,69
	7	15	2	3	3,00	50,00	0,00	0,36	3	3	50,00	2,16	2,69
MITCHELL	21	39	3	3	3,00	0,00	0,00	1,69	3	3	0,00	12,83	16,01
	21	14	8	10	10,05	25,00	0,00	1,70	9	9	12,50	12,70	15,86
	21	15	8	8	8,85	0,00	0,00	1,72	8	8	0,00	12,77	15,94
	21	21	5	6	6,50	20,00	0,00	1,75	6	6	20,00	13,00	16,22
	21	26	5	5	5,10	0,00	0,00	1,76	5	5	0,00	13,26	16,55
	21	35	3	4	4,00	33,33	0,00	1,75	4	4	33,33	13,42	16,74
SAWYER	30	30	12	14	14,55	16,67	0,00	3,76	12	12	0,00	37,33	44,05
	30	36	10	11	11,30	10,00	0,00	3,87	10	10	0,00	28,49	35,44
	30	41	8	9	9,95	12,50	0,00	3,94	9	9	12,50	28,63	35,73
	30	54	7	7	7,15	0,00	0,00	3,93	7	7	0,00	29,11	36,34
	30	75	5	5	5,00	0,00	0,00	3,95	5	5	0,00	29,56	36,91
	30	25	14	17	17,55	21,43	0,00	3,85	14	14,9	0,00	28,37	35,29
	30	27	13	15	15,70	15,38	0,00	3,85	13	13,1	0,00	33,97	40,89
TONGE	70	364	10	11	11,20	10,00	0,00	31,40	10	10	0,00	242,28	301,98
	70	410	9	9	9,90	0,00	0,00	32,80	9	9	0,00	253,89	316,80
	70	468	8	8	8,80	0,00	0,00	32,86	8	8	0,00	251,69	314,09
	70	527	7	7	7,85	0,00	0,00	32,88	7	7	0,00	252,24	314,75
	70	176	21	25	25,05	19,05	0,00	32,91	22	22	4,76	325,83	386,36
ARC	111	5755	27	32	32,65	18,52	0,00	112,49	28	28	3,70	1268,73	1484,03
	111	8847	18	20	20,75	11,11	0,00	116,64	18	18	0,00	1167,76	1391,89
	111	10027	16	17	17,50	6,25	0,00	116,93	16	16	0,00	939,72	1167,31
	111	10743	15	16	16,65	6,67	0,00	116,96	15	15,05	0,00	922,14	1148,92
	111	11378	14	15	15,35	7,14	0,00	116,74	14	14	0,00	928,69	1158,64
	111	17067	9	10	10,00	11,11	0,00	116,43	9	9	0,00	954,95	1190,38
Total			486	533	552,65				508	509,55			

Os resultados obtidos pelo método híbrido combinado com *multi-start* apresentam uma robustez ao encontrar as soluções melhores, como pode se verificar na Tabela 2, os resultados Sol* e Sol apresentam 58 de 60 soluções idênticas. Foi encontrado usando o método BRKGA *single-start* 28 de 60 soluções ótimas, enquanto o método BRKGA+CS *multi-start* encontrou 39 de 60 soluções ótimas, mostrando a superioridade de um método em relação ao outro. Note-se também a superioridade do método com *multi-start* em relação as instâncias maiores, com mais de 30 tarefas, onde se começa a notar uma maior diferença entre os resultados obtidos pelos diferentes métodos.Os tempos computacionais apresentados pelo método com *multi-start* são sempre superiores em relação ao método com *single-start*, demonstrando um maior custo computacional que a meta-heurística híbrida necessita.

6. Conclusão

Este trabalho propõe um método híbrido (BRKGA+CS) em conjunto com a técnica *multi-start* para gerar soluções robustas para o problema de balanceamento de linha de produção (SALB-1) utilizando uma heuristica de prioridade em cinco etapas para formar as soluções. O método híbrido *Clustering Search* (CS) proposto faz uso da meta-heurística algoritmo génetico de chaves aleatórias viciadas (BRKGA) para explorar o espaço de solução do problema e da meta-heurística *multi-start* para escapar de espaços soluções pouco promissores.

O CS divide as soluções geradas em grupos, para otimizar as buscas locais em apenas regiões de soluções promissoras. Combinando o CS com a técnica *multi-start* foi encontrado soluções robustas , mesmo apresentando um tempo maior comparado com as soluções encontradas com o método com apenas o BRKGA implementado.

Os resultados computacionais mostram que a utilização da combinação do CS com o BRKGA e *multi-start* é competitivo para chegar nas soluções ótimas e necessita de poucas alterações quando aplicadas em outros problemas, já que as únicas mudanças seria na decodificação de chaves aleatórias e nas buscas locais.

Um possível alvo de estudo futuro é a pesquisa de novas buscas locais para chegar em resultados mais perto do ótimo. Procurar métodos que otimizem o tempo no qual a melhor solução é encontrada, já que as soluções que o BRKGA encontrou tem uma velocidade muito maior que o método híbrido proposto.

Agradecimentos

Este trabalho é financiado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq (Processo no 482170/2013-1, 307330/2015-0, 104379/2014-6).

Referências

Baybars, I. (1986). A survey of exact algorithms for the simple assembly line balancing problem. *Management science*, 32(8):909–932.

Bean, J. C. (1994). Genetic algorithms and random keys for sequencing and optimization. *ORSA journal on computing*, 6(2):154–160.

Bock, S. e Rosenberg, O. (1998). *A new distributed fault-tolerant algorithm for the Simple Assembly Line Balancing Problem*. Springer.

Gen, M., Tsujimura, Y., e Li, Y. (1996). Fuzzy assembly line balancing using genetic algorithms. *Computers & Industrial Engineering*, 31(3):631–634.

Ghosh, S. e Gagnon, R. J. (1989). A comprehensive literature review and analysis of the design, balancing and scheduling of assembly systems. *The International Journal of Production Research*, 27(4):637–670.

Glover, F., Laguna, M., e Martí, R. (2000). Fundamentals of scatter search and path relinking. *Control and cybernetics*, 29(3):653–684.

Gonçalves, J. F. e De Almeida, J. R. (2002). A hybrid genetic algorithm for assembly line balancing. *Journal of Heuristics*, 8(6):629–642.

Gonçalves, J. F. e Resende, M. G. (2004). An evolutionary algorithm for manufacturing cell formation. *Computers & Industrial Engineering*, 47(2):247–273.

Holland, J. H. (1975). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence.* U Michigan Press.

Michel Gendreau, J.-Y. P. (2010). *Handbook of metaheuristics*, volume 146. Springer.

Oliveira, A. C. M., Chaves, A. A., e Lorena, L. A. N. (2013). Clustering search. *Pesquisa Operacional*, 33(1):105–121.

Pomari, C. Z. e Chaves, A. A. (2014). Algoritmo híbrido com cs e brkga aplicado ao problema de alocação de berços. *SBPO-Simpósio Brasileiro de Pesquisa Operacional. Salvador–BA*.

Scholl, A. e Scholl, A. (1999). *Balancing and sequencing of assembly lines*. Physica-Verlag Heidelberg.

Scholl, A. e Voß, S. (1997). Simple assembly line balancing—heuristic approaches. *Journal of Heuristics*, 2(3):217–244.

Scholl, A., Voß, S., et al. (1994). A note on fast, effective heuristics for simple assembly line balancing problems. Technical report, Darmstadt Technical University, Department of Business Administration, Economics and Law, Institute for Business Studies (BWL).

Talbot, F. B., Patterson, J. H., e Gehrlein, W. V. (1986). A comparative evaluation of heuristic line balancing techniques. *Management science*, 32(4):430–454.