ALGORITMOS MEMÉTICOS PARA O PROBLEMA DA K-DISPERSÃO DISCRETA: ESTUDO EXPERIMENTAL APLICADO A MESORREGIÃO DO OESTE POTIGUAR

Igor Wescley Silva de Freitas, Felipe Ricardo dos Santos Fernandes, Daniel Sabino Amorim de Araújo e Dario José Aloise
Programa de Pós-Graduação em Ciência da Computação - UERN/UFERSA
Mossoró, Rio Grande do Norte - Brasil
igorwescley@gmail.com, felipe.ip@hotmail.com, aloisedj@gmail.com, daniel@imd.ufrn.br

RESUMO

O problema da k-Dispersão discreta compreende em selecionar k, dentre n facilidades, de forma que a distância mínima entre qualquer par das facilidades selecionadas seja maximizada. Este problema é aplicável a situações em que as facilidades representam ameaça mútua a sistemas de franquias e serviços. Por se tratar de um problema NP-Completo, é inviável encontrar soluções de forma eficiente através do método exato. Diante desta perspectiva, este trabalho propõe solucionar o problema da k-Dispersão discreta através de Algoritmos Mémeticos, usando duas estratégias de busca local. Posteriormente, comparar estatisticamente os resultados obtidos com soluções encontradas através do Algoritmo Genético. A instância empregada compreende a Mesorregião do Oeste Potiguar, formada por 62 cidades. Os experimentos mostram uma maior eficiência das estratégias desenvolvidas, encontrando melhores soluções que o Algoritmo Genético, e com custo computacional equivalente.

PALAVRAS CHAVE. Otimização Combinatória, Pesquisa Operacional, Inteligência Artificial.

Tópicos (OC - Otimização Combinatória, MH - Metaheurísticas)

ABSTRACT

The problem of discrete k-Dispersion comprises selecting k from among n facilities so that the minimum distance between any pair of selected facilities is maximized. This problem applies to situations where facilities pose a mutual threat to franchise systems and services. Because it is an NP-Complete problem, it is not feasible to find solutions efficiently through the exact method. From this perspective, this work proposes to solve the problem of discrete k-Dispersion through Memetic Algorithms, using two local search strategies. Subsequently, to compare the results obtained with solutions found through the Genetic Algorithm. The employed entity comprises the Mesoregion of the West Potiguar, formed by 62 cities. The experiments show a greater efficiency of the strategies developed, finding better solutions than the Genetic Algorithm, and with equivalent computational cost.

KEYWORDS. Combinatorial Optimization. Operational Research. Artificial Intelligence.

Paper topics (CO - Combinatorial Optimization, MH - Metaheuristics)

https://proceedings.science/p/85391?lang=pt-br

1. Introdução

O Problema da k-Dispersão Discreta (PKD) foi introduzido por Moon e Chaudhry [1984] e consiste em, dado um conjunto de n pontos ou facilidades alocada sobre os nós de um grafo, selecionar k facilidades dentre as n possíveis, de modo que a distância mínima entre qualquer par das k facilidades selecionadas seja máxima. Basicamente, consiste em obter k pontos ou k facilidades o mais disperso possível. Exemplos de aplicações de dispersão de facilidades surgem em diversas situações como: dispersão de centros de reabilitação criminal de centros populacionais, dispersão na localização de usinas nucleares para maximizar a segurança, dispersão na localização de lojas de uma mesma franquia, etc. Erkut [1990] define o PKD como NP-Completo, provando que se pode transformá-lo em um Problema de Clique. Dessa forma é justificável o uso de métodos heurísticos para encontrar soluções viáveis para o PKD próximas do ótimo, eventualmente, ótimas.

O Algoritmo Memético (AM) é uma metaheurística evolucionária apresentada por Moscato [1989] e é considerada como sendo uma hibridização dos Algoritmos Genéticos (AG's), o qual agrega em seu funcionamento algum procedimento (não-genético) de busca local. O AM é uma estratégia de intensificação (explotação) e diversificação (exploração), com o objetivo de aperfeiçoar indivíduos no processo ou ciclo evolutivo [Boussaïd et al., 2013] [Blum e Roli, 2003]. Segundo Moscato [2001], os AM's são considerados como um processo de busca no qual a população tenta aperfeiçoar os agentes no processo de cooperação e combinação.

Diante do exposto, este trabalho tem como proposta principal adaptar e implementar duas novas estratégias baseadas em AM, com o objetivo de resolver o PKD. Os resultados obtidos serão comparados estatisticamente, utilizando o teste *t-student*, com soluções geradas por um AG tradicional. A instância considerada para os experimentos computacionais compreende o conjunto de cidades que formam a Mesorregião do Oeste do Rio Grande Norte (RN). No total, são consideradas 62 cidades, as quais são representadas por uma matriz quadrada de distâncias.

Este trabalho está organizado da seguinte forma: na segunda seção, será apresentada a formulação do problema do PKD; a terceira seção apresenta a descrição da adaptação do AM para o PKD; a quarta seção mostra a metodologia empregada; a quinta seção é dedicada as possíveis aplicações; e por fim na sexta seção, é apresentado os resultados.

2. O Problema da k-Dispersão Discreta

Consideramos o PKD como um problema de cobertura, onde sua tarefa é localizar k facilidades de um conjunto L de instalações, de modo que a distância mínima entre qualquer par das k facilidades selecionadas seja a máxima. O problema de cobertura em questão está diretamente relacionado a localização de k-facilidades, com maior dispersão possível entre si. [Sayyady e Fathi, 2016].

Formalmente, podemos descrever o PKD como um conjunto de facilidades candidatas $L = \{1, 2, ..., n\}$ e uma matriz $n \times n$ com distâncias $(d_{ij})_{i,j \in l}$ entre os pontos i e j. A tarefa de otimização é selecionar $1 < k < n$ facilidades de L, de forma que a distância mínima entre qualquer par de pontos selecionados seja maximizada [Sayah e Irnich, 2017]. A Figura 1 representa um grafo G, onde as instalações estão sobre os vértices. Uma solução para o PKD com $k = 3$ pode ser trivialmente encontrada.

https://proceedings.science/p/85391?lang=pt-br

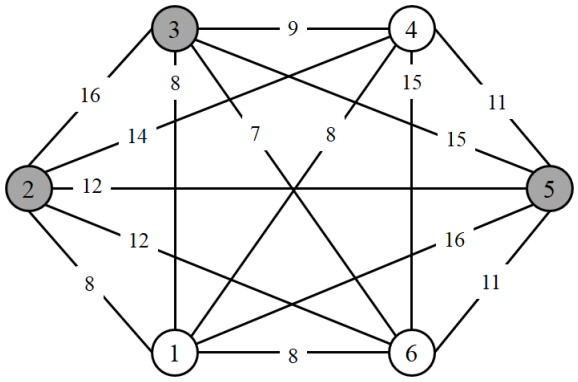


Figura 1: Possível representação de uma solução para o PKD.

A formulação matemática mais conhecida para este problema é denominada de Max- Min. A formulação que segue é devida a Erkut [1990]:

$$\text{Maximizar } w \tag{1}$$

Sujeito à:

$$w \leq d_{i,j} + M(2 - y_i - y_j), \qquad para\ todo\ i, j \in L,\ i < j \tag{2}$$

$$\sum_{j \in l} y_i = k, \tag{3}$$

$$y_i = 0\ ou\ 1,\ para\ todo\ i \in L, \tag{4}$$

onde:

w - distância mínima entre as facilidades selecionadas;

$d_{i,j}$ - distância entre as facilidades i e j;

M - número positivo representando uma distância muito maior do que $d_{i,j}$;

y_i - variável binária que assume o valor 1 se a facilidade i é selecionado e 0, no caso contrário.

A equação 1 representa a função objetivo e visa maximizar a menor distância entre os pares das facilidades. A restrição 2 se estende para cada par de pontos. Se as facilidades i e j, ambas forem selecionadas ($y_i = y_j = 1\ para\ i < j$), a restrição para (i, j) em 2 tende a formar $w \leq d_{i,j}$, de outro modo, a restrição (i, j) torna-se redundante. Consequentemente, w é igual ao mínimo das distâncias entre os pontos selecionados. A restrição 3 garante que o número de facilidades selecionadas será igual a k, e a restrição 4 trata da condição de integridade para y_i. A solução gerada é o conjunto de k-facilidades com a maior dispersão possível. [Erkut, 1990]

3. Algoritmo Memético para o PKD

Os Algoritmos Meméticos, também chamados de Algoritmos Genéticos Híbridos, permitem incorporar aos AG's procedimentos (não-genético) de refinamento que, geralmente, são técnicas de busca local. De acordo com Moscato [2001], é possível assegurar que os AM's constituem em um método mais eficaz em encontrar soluções aproximadas para problemas de otimização combinatória NP.

https://proceedings.science/p/85391?lang=pt-br

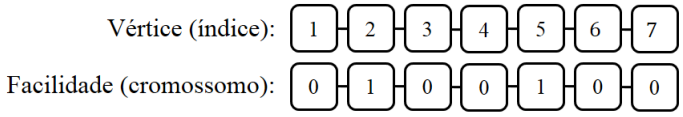


Figura 2: Representação de um indivíduo no AM adaptado ao PKD.

De forma análoga ao AG, uma solução via AM para o PKD é representada na Figura 2, onde tem-se um vetor com valores binários (Facilidades), designando ao índice do vetor (que representa os vértices) a condição de ser uma facilidade ou não.

O AM proposto considera o mesmo viés de execução de um AG tradicional, incorporando em seu ciclo evolucionário a busca local, como mostra o Algoritmo 1.

Algoritmo 1: ALGORITMO MEMÉTICO

```
1  início
2      Pop ← gerarPopulacao()
3      Pop ← avaliarFitness(Pop)
4      para i = 0, ..., maxGeracoes faça
5          Pop ← buscaLocal(Pop, probabilidade)
6          progenitores ← selecaoTorneio(Pop)
7          Pop ← cruzamento(progenitores)
8          Pop ← mutacao(Pop, probabilidade)
9          Pop ← avaliarFitness(Pop)
10         Pop ← selecao(Pop)
11     fim
12 fim
13 retorna melhor solução da Pop
```

Os mecanismos considerado pelo AM e AG são: (i) seleção por torneio, onde aleatoriamente são selecionados pares de indivíduos da população para se confrontarem ou reproduzirem; (ii) cruzamento, método de combinação do tipo corte de um ponto; (iii) mutação, método de diversificação executado aleatoriamente em indivíduos da população, de acordo com um parâmetro de probabilidade, onde os cromossomos são alterados.

Algoritmo 2: BUSCA LOCAL PARA TODA VIZINHANÇA

```
   Entrada: Pop
1  início
2      individuo ← random(Pop)
3      individuoTemp ← individuo
4      enquanto tiver vizinho faça
5          individuoNovo ← GerarVizinho(individuoTemp)
6          se individuoNovo > individuo então
7              individuo ← individuoNovo
8          fim
9      fim
10     Pop ← atualizaPop(individuo)
11 fim
12 retorna Pop;
```

https://proceedings.science/p/85391?lang=pt-br

A busca local empregada no AM segue uma característica "*Hill Climbing*", onde toda a vizinhança do indivíduo é percorrida a fim de encontrar uma *fitness* com maior representatividade. Seguindo a característica do PKD, a busca local compreende em selecionar a k facilidade, cuja aresta tem menor distância com algum par. Em seguida, busca-se outra facilidade dentre a vizinhança que derive uma melhor solução. Pode-se definir vizinhança neste problema, como qualquer facilidade ainda não selecionada, e a função $GerarVizinho()$ retorna exatamente tal facilidade de forma aleatória. O Algoritmo 2 representa o pseudocódigo da busca local mencionada.

O AG tem ordem de complexidade $O(n^4)$. Isso ocorre devido à função de *fitness* do algoritmo, que de forma independente, no pior caso tem $O(n^2)$. Com o método de torneio e o laço referente a quantidade de gerações, elevam essa complexidade. O AM que percorre toda vizinhança tem complexidade de $O(n^5)$, que se comparado com o AG, tem certa diferença no tempo de execução. Assim, propomos outro AM, este com busca local de um vizinho, tornando-o sua complexidade com polinômio equivalente a do AG, $O(3n^4)$.

Algoritmo 3: BUSCA LOCAL DE UM VIZINHO

```
Entrada: Pop
início
    individuo ← melhorFitness(Pop)
    individuoNovo ← GerarVizinho(individuo)
    se individuoNovo > individuo então
        individuo ← individuoNovo
    fim
    Pop ← atualizaPop(individuo)
fim
retorna Pop;
```

O Algoritmo 3 mostra uma outra estratégia simples de busca local, a qual considera apenas uma única solução, a partir da vizinhança da melhor solução da população, e realiza uma comparação para averiguar se esta nova solução é agora a melhor. Ambas as estratégias de busca local consideradas neste trabalho, são executadas de acordo com um parâmetro de probabilidade diretamente na população e em cada geração do algoritmo.

3.1. Trabalhos Relacionados

Lai e Hao [2016] propuseram um AM, baseado em busca tabu, para resolver o Problema da k-Dispersão de Média Máxima. A maior contribuição em seu trabalho, foi o fornecimento de um método o qual proporciona uma busca na vizinhança de forma eficiente, em tempo linear. Martí et al. [2013], compara diversas metaheurísticas, abordando o uso do AM para resolução do PKD. De acordo com Martí et al. [2013], o AM produziu bons resultados.

Em Merz e Freisleben [1999], é feito uma investigação com várias instâncias, comparando os AM's com outras metaheurísticas, tais como a Busca Tabu e Colônia de Formigas. Merz e Freisleben [1999] reportam e indicam que os melhores resultados, em todas situações, são notados aos AM's. Os AM's podem ser empregados também, em outros problemas de otimização combinatória, como por exemplo ao *job-shop scheduling*, como visto em Hasan et al. [2009], onde é proposto um AM objetivando compará-lo ao AG. Hasan et al. [2009], concluem que o AM se mostra mais eficiente em obter soluções se comparado ao AG.

4. Metodologia

Para ter dados que sustentem o bom desempenho do AM, é preciso realizar experimentos com diferentes conjuntos de parâmetros, obter as soluções e compará-las estatisticamente com soluções de outro método já conhecido na literatura. Para este trabalho, foi definido que as soluções geradas pelos AM's serão comparadas entre si, bem como com os resultados de um AG, que segue

https://proceedings.science/p/85391?lang=pt-br

o mesmo processo evolutivo do AM. Os AM's são divididos em dois: o AM 1, com busca local para todos vizinhos (conforme Algoritmo 2); e AM 2, com busca local de um vizinho (conforme Algoritmo 3).

A instância usada compreende o total de 62 cidades, que formam a mesorregião do Oeste do Rio Grande do Norte. A matriz de distâncias foi gerada usando a API do *Google Maps* [Google Maps, 2017]. O AM e o AG foram implementados utilizando a linguagem de programação Java, versão 7.

Para análise estatística dos resultados, foi utilizado o teste *t-student* bicaudal, a um nível de significância de 5%, comparando as médias de cada método e configuração. O teste *t-student* bicaudal levanta a hipótese que os dois métodos tem o mesmo desempenho, em outras palavras, que a diferença entre média dos dois métodos é igual a zero. Caso essa hipótese (hipótese nula) for rejeitada, então temos que a diferença entre os dois modelos é estatisticamente significante segundo o teste [Han et al., 2011]. Os parâmetros utilizados nos algoritmos para o experimento computacional é descrito da seguinte forma:

Tabela 1: Parâmetros e seus respectivos valores utilizados nos algoritmos.

Parâmetro	**Valor**
Tamanho da população	60
Máximo de Gerações	100; 300; 500
Número de k (facilidades)	3; 5; 10
Probabilidade de Busca Local	0.2; 0.4
Probabilidade de Cruzamento	0.5
Probabilidade de Mutação	0.2

Para cada configuração, o experimento computacional considera um total de 30 execuções consecutivas. Ao final do procedimento de execução, serão apresentados e coletados os seguintes resultados: (i) melhor solução obtida dentre as quais foram geradas; (ii) média das soluções geradas; (iii) desvio padrão; (iv) tempo computacional em milissegundos de cada execução dos algoritmos; e (v) Teste-t, um valor booleano correspondente as soluções que permite, estatisticamente, verificar a igualdade das soluções.

Os algoritmos foram executados em um notebook provido do processador Intel Core i7-CPU @2.40Hz, memória RAM instalada de 8GB e Sistema Operacional Windows 10 na plataforma 64bits.

5. Resultados

Os resultados obtidos pelo AG, AM 1 e AM 2 são mostrados na Tabela 2. A compreensão dos artefatos utilizados na Tabela 2 são entendidos da seguinte forma:

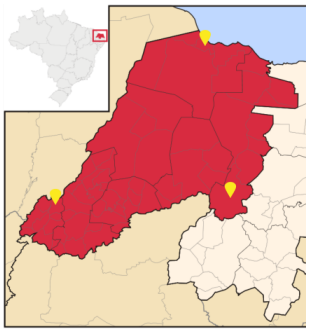
- O campo em que o resultado é apresentado em negrito, indica ser a melhor solução na configuração pertinente;
- O campo preenchido na cor cinza, indica que aquele resultado é estatisticamente igual em relação ao AG. O preenchimento também acontece no próprio AG, para melhor visualização;
- O campo sublinhado, indica a igualdade, pelo teste estatístico, dos resultados obtidos através dos AM's (AM 1 e AM 2);

Tabela 2: Resultados dos experimentos computacionais.

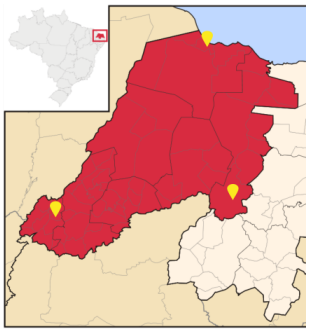
Parâmetros			**AG**				**AM 1**				**AM 2**			
Gerações	k	#BL	Solução	$\overline{x}$	σ	Tempo	Solução	$\overline{x}$	σ	Tempo	Solução	$\overline{x}$	σ	Tempo
100	3	0.2	178	146,37	16,31	310	**190**	167,37	11,71	590	**190**	157	11,85	350
		0.4	178	146,37	16,31	310	**190**	176,93	10,55	830	**190**	161,13	13,92	370
	5	0.2	94	80,63	7,47	300	115	91,67	10,93	680	**125**	90,37	9,31	490
		0.4	94	80,63	7,47	300	122	92,57	12,44	1220	**129**	90,23	9,82	390
	10	0.2	50	39,57	4,02	300	**69**	43,9	8,77	1090	61	46,53	6,12	420
		0.4	50	39,57	4,02	300	**73**	47,57	8,31	1780	69	46,8	7,36	510
300	3	0.2	170	144,5	13,4	900	**190**	167,67	15,93	1750	**190**	160,4	13,87	1090
		0.4	170	144,5	13,4	900	**190**	169,17	12,05	2540	**190**	162,77	12,12	1140
	5	0.2	98	82,57	7,92	850	105	87,73	9,01	2070	**113**	88,5	7,43	1100
		0.4	98	82,57	7,92	850	118	96,43	9,2	3160	**122**	89,77	10,43	1200
	10	0.2	52	39,8	4,43	900	59	44,7	5,76	3250	**61**	45,13	5,77	1310
		0.4	52	39,8	4,43	900	**69**	47,33	8,13	5450	67	47,33	6,39	1450
500	3	0.2	186	148,8	12,83	1510	**190**	164,13	13,22	2930	**190**	156,03	13,23	1840
		0.4	186	148,8	12,83	1510	**190**	172,5	10,32	4280	**190**	159,23	12,89	1910
	5	0.2	102	83,53	8,92	1420	115	89,97	10,84	3470	**123**	92,47	10,37	1810
		0.4	102	83,53	8,92	1420	**122**	96,2	14,55	6280	113	91,87	8,96	2020
	10	0.2	53	39,57	4,78	1450	61	44	6,82	5390	**62**	47,53	6,28	2200
		0.4	53	39,57	4,78	1450	**67**	49,43	7,01	9729	63	49,37	5,75	2430

https://proceedings.science/p/85391?lang=pt-br

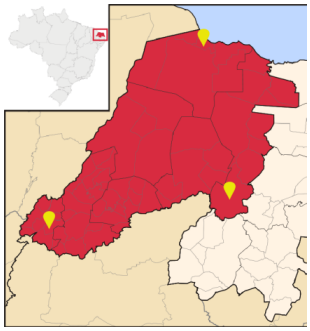
Figura 3: Representação gráfica das melhores soluções obtidas pelos AG, AM 1 e AM 2 para $k = 3$



(a) Encanto, Grossos e Jucurutu.

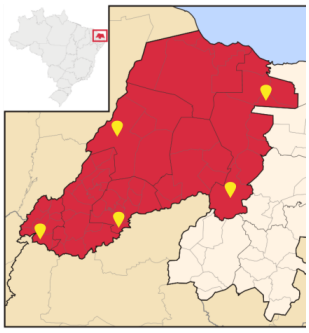


(b) Água Nova, Grossos e Jucurutu.

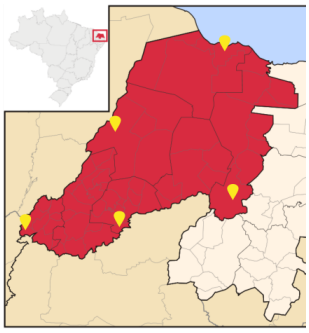


(c) Grossos, Jucurutu e Riacho de Santana.

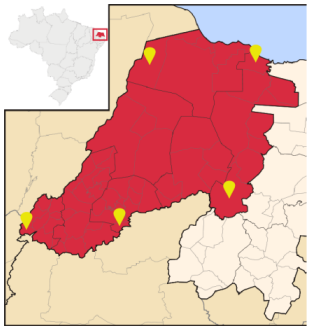
Figura 4: Representação gráfica das melhores soluções obtidas pelos AG, AM 1 e AM 2 para $k = 5$



(a) Alto do Rodrigues, Apodi, Joao Dias, Jucurutu e Luís Gomes.

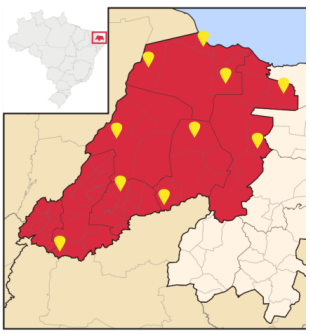


(b) Apodi, Areia Branca, Joao Dias, Jucurutu e Venha-Ver.

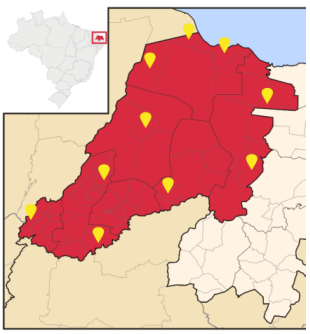


(c) Baraúna, Joao Dias, Jucurutu, Porto do Mangue e Venha-Ver.

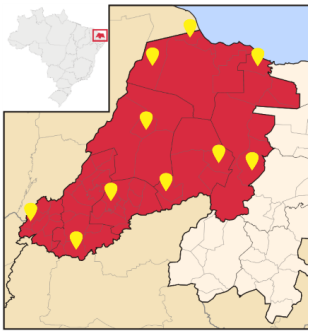
Figura 5: Representação gráfica das melhores soluções obtidas pelos AG, AM 1 e AM 2 para $k = 10$



(a) Apodi, Baraúna, Grossos, Itajá, Messias Targino, Paraná, Pendencias, Serra do Mel, Umarizal e Upanema.



(b) Alexandria, Alto do Rodrigues, Areia Branca, Baraúna, Felipe Guerra, Janduís, Riacho da Cruz, São Miguel, São Rafael e Tibau.



(c) Baraúna, Felipe Guerra, Janduís, Martins, Paraú, Porto do Mangue, São Miguel, São Rafael, Tenente Ananias e Tibau.

https://proceedings.science/p/85391?lang=pt-br

De acordo com os resultados apresentados na Tabela 2, para todas as configurações, o AM 1 encontrou melhor solução. No entanto, o AG teve um desempenho melhor em relação ao tempo de execução. A complexidade dos algoritmos explica tal comportamento. O AM 1 teve 78% (14/18) dos casos, segundo o teste-t bicaudal, estatisticamente melhor do que o AG. As melhores soluções do AG foram encontradas com 500 gerações, enquanto o AM 1 encontrou as melhores soluções para $k = 3$ e $k = 10$ com 100 gerações e para $k = 5$ com 500 gerações.

Ainda em comparação com o AG, o AM 2 encontrou melhor solução em todas as configurações. Apesar do AM 2 ter tempo de execução bem próximo do AG, ele tem desempenho de tempo inferior em todos os casos. A complexidade dos algoritmos também explica tal comportamento, pois os AM's consideram o procedimento de busca local. O AM 2 teve 61% (11/18) do casos, segundo o teste-t bicaudal, estatisticamente melhor do que o AG. O AM 2 encontrou as melhores soluções para $k = 5$ e $k = 10$ com 100 gerações e para $k = 3$ com 300 gerações.

Comparando AM 1 com AM 2, o AM 1 encontrou melhor solução em 56% (10/18) dos casos. Apesar disso, o AM 2 teve melhor tempo de execução em todas configurações. No teste-t bicaudal, 89% (16/18) dos casos são estaticamente equivalentes. Em seguida é apresentado, através das Figuras 3 (para $k = 3$), 4 (para $k = 5$) e 5 (para $k = 10$), os resultados gráficos das melhores soluções encontradas para cada estratégia.

6. Aplicações

Muitos modelos de localização ou cobertura minimizam alguma função de distância entre as facilidades. Isso é apropriado para muitos problemas que buscam encontrar instalações de serviços, onde a proximidade das facilidades é essencial. No entanto, para alguns problemas, a proximidade entre as instalações representa uma ameaça estratégica. Nesse caso, facilidades precisam estar dispersas para atingir uma ampla variedade de finalidades, seja na tentativa de otimizar vantagens competitivas do mercado, dispersar impactos negativos e até mesmo para otimizar a segurança.[Erkut e Neuman, 1989] [Curtin e Church, 2006]

A Mesorregião do Oeste Potiguar (MOP) é uma das quatro mesorregiões do estado do Rio Grande do Norte. Ocupa cerca de 40% do território estadual, compreendendo uma área de $52.796,791 km^2$. É a mesorregião com maior número de microrregiões (7) e municípios (62) [do Brasil, 2008]. Com essa grande notoriedade territorial, a MOP constitui uma área além de estratégica, desafiadora para o RN. Encontrar melhor dispersão para instalações para a MOP configura-se uma enorme contribuição para o serviço público, podendo ser aplicada na solução de localização de penitenciárias e hospitais regionais.

O isolamento através da distância, a disponibilidade de terrenos, menor resistência da população e dos governantes tem sido os principais fatores de decisão quando se planeja instalação de penitenciárias no Brasil [Júnior, 2014]. Todavia, o principal aspecto que deve ser levado em consideração quando se planeja uma unidade penitenciária é a dispersão [Melossi e Pavarini, 2018]. Isso em razão de cobertura, para otimizar a logística envolvida, tanto para o sistema de segurança quanto para os familiares dos detentos; E dispersão de índices criminais ocasionados ou ampliados devido com a instalação da unidade. Esse problema de localização, se adequa-se plenamente com o PKD. As estratégias meméticas propostas neste trabalho podem ser aplicadas de forma eficiente, na localização dispersa de unidades penitenciárias na MOP.

De forma análoga ao problema de localização de penitenciárias, regionalização de hospitais públicos pode ser tratado como um PKD. Nos últimos anos, propostas para regionalização de serviços públicos de saúde vem sendo discutidas, a fim de atender a população com maior qualidade de serviço e menor custo. A regionalização envolve dispersar esses serviços, de maneira que tenha-se uma maior cobertura territorial [Galvão et al., 1999]. Localizar os municípios mais dispersos para implantação de hospitais regionais, também caracteriza-se como uma possível aplicação para as estratégias propostas neste trabalho.

https://proceedings.science/p/85391?lang=pt-br

7. Considerações Finais

Os Algoritmos Meméticos trazem uma contribuição valorosa para soluções em problemas de otimização combinatória. A busca local aliada ao processo evolutivo intuitivamente leva ao encontro de melhores soluções. Para Moscato [2001], os AM's estão demonstrando resultados impressionantes em problemas práticos de otimização. Existem experiências surpreendentes em diversas áreas, tais como medicina, oceanografia e economia. Este fato demonstra que a hibridização de busca local com algoritmos genéticos produz efeitos práticos na descoberta de melhores soluções em tempo polinomial.

O AM proposto neste trabalho teve um papel importante na busca de melhores soluções para o PKD. Os resultados mostram que as duas estratégias de AM implementadas encontram soluções superiores ao AG em todas as configurações. No teste *t-student*, os resultados seguem com o mesmo viés, sendo que o AM 1 (Algoritmo 2) teve melhor desempenho. A estratégia de busca local para toda vizinhança, apesar de ter maior ordem de complexidade, garante ao AM 1 superioridade no teste estatístico e em melhores soluções encontradas. O AG por sua vez, conta com o tempo de execução menor comparado com os AM's, sustentado por uma ordem de complexidade inferior ao AM. Contudo, o AM teve desempenho de tempo equivalente ao AG, devido a sua complexidade de mesmo grau polinomial, e desempenho qualitativo semelhante ao AM (Algoritmo 2), devido a sua Busca Local de um vizinho. Por esse aspecto, a estratégia empregada no AM (Algoritmo 3) pode ser considerada mais eficiente. Importante observar também, que apesar de que os AM's tenham maior tempo de execução, suas melhores soluções foram encontradas com 100 e 300 gerações, enquanto o AG encontrou suas melhores soluções somente com 500 gerações.

A contribuição do presente trabalho está na implementação de duas estratégias de AM para o PKD, com resultados satisfatórios, demonstrando maior qualidade nas soluções, eficiência equivalente, ou melhor. É importante em trabalhos futuros, provar a melhor eficiência dos AM's quando encontram melhores soluções com menos gerações. Para este, foi demonstrado que para o PKD o AM representa uma metahuerística eficiente se comparado com o AG.

Referências

Blum, C. e Roli, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Comput. Surv.*, 35(3):268–308. ISSN 0360-0300. URL http://doi.acm.org/10.1145/937503.937505.

Boussaïd, I., Lepagnot, J., e Siarry, P. (2013). A survey on optimization metaheuristics. *Information Sciences*, 237(Supplement C):82 – 117. ISSN 0020-0255. URL http://www.sciencedirect.com/science/article/pii/S0020025513001588. Prediction, Control and Diagnosis using Advanced Neural Computations.

Curtin, K. M. e Church, R. L. (2006). A family of location models for multiple-type discrete dispersion. *Geographical Analysis*, 38(3):248–270.

do Brasil, D. T. (2008). Divisão territorial do brasil e limites territoriais. *Instituto Brasileiro de Geografia e Estatística (IBGE)*, 11.

Erkut, E. (1990). The discrete p-dispersion problem. *European Journal of Operational Research*, 46(1):48 – 60. ISSN 0377-2217. URL http://www.sciencedirect.com/science/article/pii/037722179090297O.

Erkut, E. e Neuman, S. (1989). Analytical models for locating undesirable facilities. *European Journal of Operational Research*, 40(3):275–291.

Galvão, R. D., Nobre, F. F., e Vasconcellos, M. M. (1999). Modelos matemáticos de localização aplicados á organização espacial de unidades de saúde. *Revista de Saúde Pública*, 33(4):422–434.

https://proceedings.science/p/85391?lang=pt-br

Google Maps (2017). Google maps apis. http://www.google.cn/apis/maps/signup.html. 16 de nov. de 2017.

Han, J., Pei, J., e Kamber, M. (2011). *Data mining: concepts and techniques*. Elsevier.

Hasan, S. M. K., Sarker, R., Essam, D., e Cornforth, D. (2009). Memetic algorithms for solving job-shop scheduling problems. *Memetic Computing*, 1(1):69–83. ISSN 1865-9292. URL https://doi.org/10.1007/s12293-008-0004-5.

Júnior, J. H. Z. (2014). cartografias da interiorização penitenciária no estado de são paulo. *verve. revista semestral autogestionária do Nu-Sol.*, (25).

Lai, X. e Hao, J.-K. (2016). A tabu search based memetic algorithm for the max-mean dispersion problem. *Computers & Operations Research*, 72:118 – 127. ISSN 0305-0548. URL http://www.sciencedirect.com/science/article/pii/S0305054816300405.

Martí, R., Gallego, M., Duarte, A., e Pardo, E. G. (2013). Heuristics and metaheuristics for the maximum diversity problem. *Journal of Heuristics*, 19(4):591–615. ISSN 1572-9397. URL https://doi.org/10.1007/s10732-011-9172-4.

Melossi, D. e Pavarini, M. (2018). *The prison and the factory: Origins of the penitentiary system*. Springer.

Merz, P. e Freisleben, B. (1999). A comparison of memetic algorithms, tabu search, and ant colonies for the quadratic assignment problem. In *Proceedings of the 1999 Congress on Evolutionary Computation (CEC)*, volume 3, p. 2070.

Moon, I. D. e Chaudhry, S. S. (1984). An analysis of network location problems with distance constraints. *Management Science*, 30(3):290–307. URL https://doi.org/10.1287/mnsc.30.3.290.

Moscato, P. (1989). On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. *Caltech concurrent computation program, C3P Report*, 826:1989.

Moscato, P. (2001). Problemas de otimização np, aproximabilidade e computação evolutiva: Da prática à teoria.

Sayah, D. e Irnich, S. (2017). A new compact formulation for the discrete p-dispersion problem. *European Journal of Operational Research*, 256(1):62 – 67. ISSN 0377-2217. URL http://www.sciencedirect.com/science/article/pii/S037722171630457X.

Sayyady, F. e Fathi, Y. (2016). An integer programming approach for solving the p-dispersion problem. *European Journal of Operational Research*, 253(1):216–225.

https://proceedings.science/p/85391?lang=pt-br