XIII Encontro de Modelagem Computacional
Instituto Politécnico (IPRJ), Campus Regional da UERJ, Nova Friburgo/RJ, Brasil. 03-05 nov. 2010
Associação Brasileira de Engenharia e Ciências Mecânicas - ABCM

COMPARAÇÃO DE HEURÍSTICAS APLICADAS AO PROBLEMA DA DIVERSIDADE MÁXIMA

FREITAS, A. A. R.- alandefreitas@gmail.com
Universidade Federal do Ouro Preto, Departamento de Ciência da Computação
35400-000 - Ouro Preto, MG, Brasil
SOUZA, M. J. F.- marcone@iceb.ufop.br
GUIMARÃES, F. G.- fredericoguimaraes@ufmg.br
Universidade Federal de Minas Gerais, Departamento de Engenharia Elétrica
31270-010 - Belo Horizonte, MG, Brasil

Abstract. *O Problema da Diversidade Máxima consiste em escolher um subconjunto de elementos que tenham diversidade máxima entre si. A diversidade entre dois elementos pode ser apresentada de várias maneiras e podem ser salvas em uma matriz de diversidade. Já a procura do conjunto mais diverso possível é um problema NP-Completo, o que justifica a utilização de heurísticas. Neste artigo são apresentadas algumas abordagens heurísticas, que produzem bons resultados para o problema, com relação aos melhores resultados da literatura. A heurística baseada em estratégias evolutivas desenvolvida neste trabalho apresentou os melhores resultados.*

Keywords: *Problema da Diversidade Máxima, Iterated Local Search, Estratégias Evolutivas*

A COMPARISON OF HEURISTICS APPLIED TO THE MAXIMUM DIVERSITY PROBLEM

Abstract. *The Maximum Diversity Problem consists in choosing a subset of elements with maximum diversity. The diversity between two elements can be given by many problem-specific ways and can be stored in a diversity matrix. The search for the most diverse set is an NP-Complete problem, justifying the use of heuristics. In this paper, some heuristic approaches that provide good results for the problem are presented, considering the best results known in literature. The heuristic method based on evolution strategies, developed in this work, achieved the best results.*

Keywords: *Maximum Diversity Problem, Iterated Local Search, Evolution Strategies*

1. INTRODUÇÃO

O Problema da Diversidade Máxima (PDM) consiste em escolher, dentre um conjunto N com n elementos, um subconjunto M de m elementos que gere a diversidade máxima entre si. A diversidade entre cada par de elementos $i, j \in M$ é calculada por uma função d_{ij}. A qualidade da solução pode ser avaliada pelo somatório de todos os valores de d_{ij} possíveis. Neste trabalho, é resolvido o problema de maximização da diversidade. Usando-se a mesma representação, pode-se resolver também o problema de procurar a menor diversidade entre os elementos do conjunto.

O problema é NP-difícil e sua redução para o problema de clique em grafos pode ser encontrada em (Kuo et al., 1993). A partir de uma redução ao problema de recobrimento de vértices, foi também provado que tanto o algoritmo para maximizar quanto para minimizar a diversidade de um subconjunto de tamanho m é NP-Difícil (Ghosh, 1996). Em vista disso, não se conhecem métodos matemáticos polinomiais para a resolução deste problema. Dada essa dificuldade de solução do PDM, neste trabalho serão desenvolvidas algumas heurísticas para sua solução e feita uma comparação entre elas.

2. DEFINIÇÃO DO PROBLEMA

Neste problema, tem-se um conjunto de certos valores d_{ij}, armazenados em uma matriz $D[n \times n]$, que representam a diversidade entre os elementos i e j pertencentes a um conjunto N de tamanho n. Deve-se encontrar um subconjunto M com $m < n$ elementos de diversidade máxima. A diversidade deste subconjunto M é medida pela soma das diversidades d_{ij} entre todos os seus elementos. Quanto mais o valor de m se aproxima de $n/2$, maior é o espaço de busca do problema. Em cada área do conhecimento, esta medida de diversidade pode ser feita de uma forma diferente. O problema pode ser definido como:

$$\max \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} d_{ij} x_i x_j, \text{ sujeito a: } \sum_{i=1}^{n} x_i = m, \; x_i \in \{0,1\} \, \forall i = 1, \cdots, n. \tag{1}$$

3. ESTADO DA ARTE

Abordagens anteriores para solucionar o PDM incluem um *Greedy Randomized Adaptative Search Procedure* (GRASP) (Ghosh, 1996), com boa eficiência para instâncias em que $n \leq 40$, e *Simulated Annealing* (Agrafiotis, 1997), com resultados satisfatórios de acordo com o autor. Em (Cutler, 1997) foram desenvolvidos um método baseado em relaxação lagrangeana e uma heurística gulosa que, quando comparados a um algoritmo *branch-and-bound*, produziram bons resultados para instâncias em que $n \leq 40$. Em um estudo sobre a criação de grupos de projeto em classes de estudantes (Weitz & Lakshminarayanan, 1998), foram apresentadas cinco heurísticas para o problema. Todas elas foram testadas usando uma instância feita de dados reais e a heurística *Lotfi-Cerveny-Weitz* foi apontada como a melhor.

Em (Glover et al., 1998) foram desenvolvidas heurísticas construtivas que produzem soluções a apenas 2% do valor ótimo global, no máximo. Também foi testada uma Busca Tabu em instâncias envolvendo de 100 a 1000 elementos e valores de m de 10% a 30% de n. Os resultados, porém, não foram comparados a outros da literatura.

Os métodos apresentados anteriormente foram melhorados com aplicações de GRASP com diferentes heurísticas construtivas e de busca local (Silva et al., 2004). Essa abordagem foi mais tarde aperfeiçoada com um módulo de reconexão por caminhos (Silva et al, 2007).

Os trabalhos mais recentes destinados a resolver este problema incluem algoritmos *branch-and-bound* (Martí et al., 2010), *Scatter Search* com Reconexão por Caminhos (Resende et al., 2010) e GRASP com Reconexão por Caminhos (Resende et al., 2010). Os melhores algoritmos conhecidos para a resolução do PDM na literatura são aplicações do algoritmo Tabu_D2 (Duarte & Martí, 2007) e uma versão do *Iterated Local Search* (ILS) (Duarte et al., 2008).

4. *ITERATED LOCAL SEARCH*

O *Iterated Local Search* (ILS) é uma heurística na qual novas soluções são obtidas a partir de perturbações em ótimos locais encontrados durante a busca. Para comparações neste trabalho, foram feitos testes com uma implementação de ILS (Duarte et al., 2008) que gerou os

melhores resultados conhecidos na literatura. O pseudocódigo do ILS encontra-se disponível em (Lourenço et al, 2003).

Uma heurística gulosa gera a solução inicial. Para cada elemento i de N, são escolhidos os m elementos que têm maior diversidade em relação ao elemento i. A soma das diversidades entre o elemento i e cada elemento dos m escolhidos é guardada em s_i. No fim, os m elementos com maior s_i são selecionados.

A perturbação de uma solução é feita trocando um de seus elementos, definido aleatoriamente, por outro elemento não presente na solução. Existem 7 níveis de perturbação, sendo que nos seis primeiros são aplicadas, respectivamente, 2%, 10%, 40%, 50%, 70% e 80% de m modificações nos elementos da solução. O sétimo nível consiste em alterar apenas um elemento da solução, independentemente do tamanho do problema, sendo aplicado para alterações mais suaves na solução.

Neste trabalho, considera-se como critério de aceitação que uma solução $s^{*\prime}$ passa a ser a solução corrente apenas se ela melhorar a solução global, isto é, se $f(s^{*\prime}) > f(s^*)$.

Os métodos de busca local foram definidos baseando-se na ideia de que um vizinho de uma solução s é uma solução s' qualquer que pode ser obtida a partir da troca de um elemento pertencente a M na solução s com um elemento pertencente a $N - M$ na mesma solução s. Os quatro tipos de métodos utilizados como busca local estão definidos nas próximas subseções.

Melhor Vizinho Cada elemento i pertencente à solução é trocado por um elemento j não pertencente a ela. A troca que leva a uma maior melhoria na função objetivo é escolhida para ser de fato aplicada à solução corrente e uma nova iteração do algoritmo ocorre. Como o melhor vizinho é sempre escolhido, sempre que se executa o algoritmo para as mesmas entradas, tem-se a mesma saída. Esta busca local é conhecida como busca *Best Improvement* (BI).

Primeiro de Melhora A busca local de primeiro de melhora se assemelha muito à busca local de melhor vizinho. Porém, nesse caso, a primeira solução encontrada que melhora a função objetivo é usada como solução da próxima iteração e a iteração corrente é encerrada. A ordem na qual os vizinhos serão testados pode ser aleatória, o que pode levar a diferentes respostas mesmo quando se tem as mesmas entradas para os algoritmos. Esta busca local é conhecida como busca *First Improvement* (FI).

Busca Randômica Na busca randômica, um vizinho qualquer de uma solução s é escolhido aleatoriamente. Se este vizinho melhora a função objetivo, ele passar a ser a solução corrente. Após isso, inicia-se uma nova iteração e volta-se a gerar um novo vizinho, qualquer que seja a solução corrente. O algoritmo encerra quando um critério de parada é atingido. Neste trabalho, o número de iterações sem melhora foi usado para definir este critério.

Primeiro de Melhora ou Randômica A quarta alternativa de busca local considerada neste trabalho consiste em decidir de maneira aleatória entre a busca local de primeira melhora e a busca randômica. Essa estratégia foi adotada para que se possa explorar mais o espaço de busca com a busca randômica, mas também intensificar a exploração do espaço de soluções com a busca FI.

5. ESTRATÉGIA EVOLUTIVA

Para comparação com os algoritmos já existentes na literatura, foi implementada uma Estratégia Evolutiva (EE), um algoritmo populacional baseado em mutações. EE mais simples são as $(1 + 1)$ baseadas na mutação de um único indivíduo para gerar um único descendente por geração. Existem também a estratégia $(\mu + 1)$, na qual uma população de μ indivíduos se

combina para gerar apenas um indivíduo que pode substituir o pior elemento da população, e a estratégia $(\mu + \lambda)$ e (μ, λ), em que existem λ descendentes que competem com os pais na primeira abordagem ou os substituem na segunda.

Neste trabalho foi usada a abordagem EE(μ, λ) para aumentar a diversidade da busca removendo todos os pais a cada geração. Assim, dada uma população de μ indivíduos, cada um com n genes, o genótipo dos descendentes terá sempre uma pequena diferença em relação ao genótipo dos pais.

Como cada indivíduo produzirá, na média, λ/μ descendentes, então existirão λ novos indivíduos. Dentre esses, apenas μ sobreviverão para formar a próxima geração. O pseudocódigo da EE é apresentado no Algoritmo 1.

Algoritmo 1: Estratégia Evolutiva

Entrada: Problema
Resultado: Solução para o Problema
$X_t \leftarrow$ população inicial;
Inicializar os parâmetros da estratégia;
enquanto *critério de parada não é satisfeito* **faça**
  para i *de* 1 *até* λ **faça**
    Escolha ρ pais aleatoriamente;
    Faça recombinação para gerar gerar z;
    Faça mutação nos parâmetros da estratégia;
    $z \leftarrow z$ + mutação;
    Avaliar $f(z)$;
  fim
  Aplicar seleção $EE(\mu/\rho, \lambda)$;
  $t \leftarrow t + 1$;
fim
retorna *a melhor solução visitada*;

Pode-se perceber que existe uma grande semelhança do EE(1+1) com o ILS. Nos dois métodos uma solução s^* é perturbada e a melhor delas passa a ser a solução corrente. Porém, quando é feita a utilização de EE(μ, λ), tem-se algumas potenciais vantagens de métodos populacionais, pois o resultado de diferentes mutações pode dar informações sobre o melhor modo de se guiar o processo.

Neste trabalho é também usada auto-adaptação, ou seja, a informação sobre como deve ocorrer a mutação dos indivíduos está codificada no genótipo do próprio indivíduo. Espera-se assim que indivíduos que tenham bons parâmetros de mutação para o problema sobrevivam mais vezes e esses valores mais favoráveis sejam usados em gerações futuras.

5.1 Parâmetros do Algoritmo

Neste trabalho foram feitos testes com $\mu = 10$ e $\lambda = 50$. Todos os indivíduos são inicializados com uma solução gerada por uma heurística construtiva como a descrita na seção 4., dado que no vasto espaço de busca do problema, partir de soluções aleatórias seria um processo de convergênca muito lento. Mesmo com o método construtivo determinístico como o utilizado neste trabalho, a EE é um método populacional em que é aceitável a inicialização de indivíduos de modo que levem a uma menor diversidade na população se estes tem uma boa qualidade já inicialmente. Isso acontece porque a EE é um método fortemente baseado em mutação.

5.2 Recombinação

Como a recombinação de pais pode ser considerada uma operação de custo computacional relativamente alto para o PDM – já que para o PDM é possível calcular a diferença da qualidade entre duas soluções sem se ter que necessariamente calcular a qualidade destas duas soluções –, a recombinação entre dois pais acontece com uma chance de 0,5% na geração de um filho. Os outros filhos são baseados no material genético de apenas um pai. Este valor de probabilidade de cruzamento poderia ser considerado péssimo para um Algoritmo Genético (AG) tradicional, porém é um valor que funciona bem para EE e para o PDM.

A recombinação é feita com a junção entre todos os elementos i que pertencem ao mesmo tempo à solução representada pelo pai_1 e à solução do pai_2. Caso o descendente não venha do cruzamento de dois pais, ele será inicialmente apenas a cópia de um pai qualquer selecionado aleatoriamente.

5.3 Auto-adaptação

Após a etapa de recombinação, é feita a mutação nos parâmetros da estratégia antes de se aplicar a mutação nos próprios indivíduos. Para cada indivíduo existem 7 valores que são definidos aleatoriamente entre 0 e 1 na inicialização do algoritmo. Estes valores definem a chance de cada mutação ser aplicada no indivíduo. De acordo com estas probabilidades, escolhe-se apenas uma mutação para ser aplicada sobre o indivíduo.

As mutações funcionam de maneira semelhante à perturbação de solução descrita na seção 4.. Uma certa porcentagem de indivíduos pertencentes à solução é trocada com elementos não pertencentes à mesma. Os valores possíveis para perturbação são novamente 2%, 10%, 40%, 50%, 70%, 80% do número de elementos da solução e um sétimo nível que troca apenas um elemento da solução independentemente do tamanho do problema. Porém, na EE, as probabilidades de aplicação de cada uma destas mutações é uma característica do indivíduo, enquanto no ILS, esta era sempre de $1/7$.

Quando ocorre o cruzamento entre dois pais, os parâmetros de mutação σ sofrem mutação tendo-se como referência os valores dos pais. Os parâmetros σ_z do filho são definidos por $\alpha\sigma_{pai1} + (1-\alpha)\sigma_{pai2}$, sendo α um número aleatório entre 0,3 e 0,7.

Os parâmetros da estratégia sofrem uma mutação normal. Cada taxa de probabilidade σ_i de um indivíduo é acrescida de um valor Δ que é o próprio σ_i multiplicado por um valor aleatório definido pela normal que por sua vez é multiplicado por 0,2. Assim, a mutação terá uma abrangência de no máximo 20% do indivíduo. Além disto, nenhum dos valores de σ_i pode ultrapassar $(4/7)\sum_{i=1}^{7}\sigma_i$.

5.4 Mutação

Com a auto-adaptação, cada indivíduo tem 7 valores de σ_i que representam a probabilidade de cada uma das 7 mutações de ocorrer. O método da roleta é utilizado para fazer isto: é sorteado um número k de 0 até $\sum_{i=1}^{7}\sigma_i$. É criado um loop com i variando de 1 até 7 em que os valores de σ_i são adicionados a uma variável $soma$ que é inicializada com 0. Quando houver um primeiro valor de σ_i que faça com que $soma$ seja maior que k, o *loop* é encerrado e o último valor i no *loop* é a posição da mutação escolhida. Com este método, cada mutação i que tem sua probabilidade de ocorrência representada por σ_i tem chance $\frac{\sigma_i}{\sum_{j=1}^{7}\sigma_j}$ de ocorrer.

5.5 Avaliação e Seleção dos Indivíduos

Na implementação da EE proposta neste trabalho, após a mutação, todos os indivíduos passam por uma busca local randômica como a descrita na seção 4. antes de ser avaliada. Isso

ocorre pois uma solução que sofre uma perturbação ou mutação sem passar por um refinamento não teria muita chance de competir com outros indivíduos, no caso de comparações entre indivíduos que sofreram mutações maiores e outros que sofreram perturbações menores. Isso ocorre pois os filhos, após algumas gerações, serão descendentes de pais de boa qualidade. Para poder distinguir melhor entre os indivíduos e escolher os λ melhores para compor a nova solução, é aplicada uma busca randômica entre todos os indivíduos. A busca randômica é escolhida pois existem vários indivíduos λ dos quais apenas μ serão escolhidos, o que gera um *trade-off* entre qualidade da avaliação e tempo de execução.

Somente após este refinamento com a busca randômica, as μ melhores soluções entre os descendentes são passadas para a nova geração. Antes de se iniciar a nova geração, todos os μ indivíduos selecionados passam por uma busca FI como a descrita na seção 4.. Assim, soluciona-se de certa maneira o problema do tempo computacional, pois apenas os indivíduos selecionados passam pela busca FI. Além disso, tem-se também uma referência para escolher entre os indivíduos, pois estes passaram por uma busca randômica.

6. RESULTADOS

Para efeito de comparação, foram utilizadas instâncias já conhecidas na literatura (Silva et al., 2004) para comparar algoritmos para o PDM. As instâncias têm tamanho $n \in \{100, 200, 300, 400, 500\}$ e tamanhos de $m \in \{0.1n, 0.2n, 0.3n, 0.4n\}$. Os valores de diversidade $d(i, j)$ são números aleatórios entre 0 e 9.

Para os testes, cada heurística foi executada 10 vezes para cada instância com um tempo de execução máximo de 5 minutos cada. O critério de parada era encontrar o melhor valor conhecido na literatura (Martí, 2008) para o problema ou alcançar o tempo de 5 minutos. Foram comparadas as quatro abordagens de ILS apresentadas na seção 4. (com Busca BI, Busca FI, Busca Randômica e escolha aleatória entre Busca FI e Busca Randômica) e a abordagem de EE apresentada na seção 5..

Na Tabela 1 estão os resultados destes testes para as instâncias de 1 a 10 e 11 a 20. Nas segunda e terceira linhas estão as dimensões das instâncias. Na quarta linha (*Best*) está o melhor valor conhecido para a instância. Nas outras linhas estão os resultados médios obtidos com cada algoritmo com desvio-padrão, o tempo médio gasto em minutos (Min.) e o desvio-padrão do tempo gasto. O critério de parada dos algoritmos era encontrar o melhor valor conhecido ou atingir o limite de tempo de 5 minutos. Os algoritmos que o limite estão com o símbolo 5+.

Como as instâncias foram todas definidas com valores de $d(i, j)$ que variam de 0 a 9, é importante mencionar certas propriedades do problema. Como os valores escolhidos para $d(i, j)$ não variam em função de n, pode ser mais fácil encontrar valores próximos do ótimo quando o valor de n é mais alto. Isso acontece pois quando n aumenta, os valores de $d(i, j)$ ficam menos relevantes em relação ao tamanho total da instância. Assim, algoritmos simples podem levar a valores mais próximos ao ótimo quando n aumenta. Demonstrando essa propriedade, na Tabela 2 estão os valores encontrados pela heurística construtiva e os melhores da literatura.

Pelos resultados da Tabela 2 pode-se perceber que quanto maior o valor de n e quanto mais próximo de $n/2$ é o valor de m – que são os problemas mais difíceis de serem resolvidos –, mais próxima da qualidade da melhor solução é a solução gerada pela heurística construtiva. Isso acontece pois os valores $d(i, j)$ não foram definidos em função de n e pode levar a acreditar que algoritmos mais simples são melhores que outros mais robustos, especialmente em comparações em que se tem como referência uma meta em relação ao ótimo a ser atingida.

Com resultados próximos aos das melhores soluções encontrados pela heurística construtiva, métodos de busca menos sofisticados podem ser eficientes quando comparados a algoritmos mais complexos em testes rápidos, sem tempo para explorar bem o espaço de busca.

Table 1: Resultados Obtidos em um Tempo Máximo de 5 minutos

Inst.	1	2	3	4	5	6	7	8	9	10
n	100	100	100	100	200	200	200	200	300	300
m	10	20	30	40	20	40	60	80	30	60
Best	333	1195	2457	4142	1247	4450	9437	16225	2694	9689
ILS 1	333	1195	2457	4141.2	1245.8	4441.1	9395	16210	2675.9	9624.3
std	0	0	0	1.687	1.932	5.744	16.25	0	7.68	25.97
Min.	0.271	0.843	0.0147	1.99	2.64	5+	5+	5+	5+	5+
std	0.216	0.686	2×10^{-5}	2.03	1.88	0	0	0	0	0
ILS 2	333	1195	2457	4142	1244.6	4445.2	9422.9	16215.7	2685.4	9658.5
std	0	0	0	0	2.066	3.225	8.569	7.761	5.835	20.58
Min.	0.0981	0.227	0.324	0.511	3.37	4.63	4.59	4.71	4.81	5+
std	0.0891	0.197	0.882	0.658	2.15	1.2	1.49	1.15	0.687	0
ILS 3	333	1195	2457	4142	1245.2	4444.8	9428	16222.1	2679	9657.4
std	0	0	0	0	2.394	3.824	11.47	5.934	5.676	16.35
Min.	0.0851	0.112	0.0187	2.03	2.46	4.51	2.84	3.31	5+	5+
std	0.075	0.0769	0.0291	1.68	2.27	1.55	2.33	2.29	0	0
ILS 4	333	1195	2457	4142	1245.4	4444.7	9424.2	16220.6	2685.6	9654.5
std	0	0	0	0	2.066	3.773	9.75	7.933	6.022	17.68
Min.	0.0508	0.205	0.49	0.213	2.71	4.63	4.49	4.12	5+	5+
std	0.0471	0.275	0.838	0.159	2.31	1.27	1.53	2.07	0	0
EE	333	1195	2457	4142	1247	4448.9	9436.2	16225	2688.4	9688.7
std	0	0	0	0	0	2.424	2.53	0	5.42	0.483
Min.	0.0334	0.0486	0.0561	0.0728	0.423	1.49	1.27	0.462	3.87	2.8
std	0.0108	0.0164	0.00403	0.0324	0.738	1.88	1.6	0.274	1.85	1.64

Inst.	11	12	13	14	15	16	17	18	19	20
n	300	300	400	400	400	400	500	500	500	500
m	90	120	40	80	120	160	50	100	150	200
Best	20743	35881	4658	16956	36317	62487	7141	26258	56572	97344
ILS 1	20641.6	35836.3	4616.6	16836	36200	62254.1	7049	26103.2	56407	97194
std	36.86	20.72	14.54	30.64	0	79.77	20.12	58.89	0	0
Min.	5+	5+	5+	5+	5+	5+	5+	5+	5+	5+
std	0	0	0	0	0	0	0	0	0	0
ILS 2	20699	35869.4	4622.3	16886.9	36205.2	62426	7092.8	26115.3	56551	97232.5
std	0	0.8433	11.55	24.69	28.28	0	22.92	16.01	0	37.89
Min.	5+	5+	5+	5+	5+	5+	5+	5+	5+	5+
std	0	0	0	0	0	0	0	0	0	0
ILS 3	20710.9	35855.2	4639.8	16901.9	36247.9	62403.9	7097.6	26191.6	56499.4	97260.4
std	14.84	13.31	12.41	14.11	23.59	20.36	16.78	35.63	40.36	61.55
Min.	5+	4.76	5+	5+	5+	5+	5+	4.66	5+	4.98
std	0	0.7691	0	0	0	0	0	1.0995	0	0.088011
ILS 4	20702.4	35873.7	4633.3	16876.4	36201.1	62412.6	7092.2	26153.2	56526.4	97191
std	8.099	5.736	13.86	37.08	38.29	42.37	22.32	45.99	54.79	78.95
Min.	5+	4.6	5+	5+	5+	5+	5+	5+	5+	5+
std	0	1.6584	0	0	0	0	0	0	0	0
EE	20739.3	35878.6	4653.4	16936.1	36302.3	62467.3	7127	26250.3	56571.7	97326.1
std	6.929	2.633	6.363	10.71	8.125	18.71	11.85	10.81	0.6749	13.54
Min.	2.58	4.51	3.47	4.81	5+	5+	4.75	4.09	4.62	5+
std	1.9261	1.3537	2.0576	0.98106	0	0	0.9662	1.2971	0.85154	0

Table 2: Resultados Obtidos pela Heurística Construtiva

Instância	n	m	Heurística Construtiva	*Best*	Desvio
1	100	10	216	333	35.1351%
2	100	20	1003	1195	16.0669%
3	100	30	2099	2457	14.5706%
4	100	40	3852	4142	7.0014%
5	200	20	883	1247	29.1901%
6	200	40	3836	4450	13.7978%
7	200	60	8526	9437	9.6535%
8	200	80	15434	16225	4.8752%
9	300	30	1899	2694	29.51%
10	300	60	8305	9689	14.2842%
11	300	90	19002	20743	8.3932%
12	300	120	34220	35881	4.6292%
13	400	40	3540	4658	24.0017%
14	400	80	14623	16956	13.7591%
15	400	120	33998	36317	6.3854%
16	400	160	59697	62487	4.4649%
17	500	50	5490	7141	23.12%
18	500	100	23452	26258	10.6863%
19	500	150	53346	56572	5.7025%
20	500	200	93835	97344	3.6047%

Porém, mesmo quando é mais fácil encontrar soluções de qualidade próximas à do ótimo, o espaço de busca do problema não diminui. Sempre existirão $n!/(m!(n-m)!$ soluções possíveis e apesar de ser mais fácil de se achar soluções com qualidade próxima ao ótimo, a dificuldade para alcançar o ótimo do problema continua a mesma.

Por isso, foram feitos testes tendo o tempo necessário para encontrar o melhor valor conhecido como critério de parada, o que é um melhor critério neste caso.

Na Tabela 3 é mostrado o tempo gasto (em minutos) por cada um dos algoritmos para encontrar o melhor valor conhecido. Estão com o símbolo 5+ as situações em que o algoritmo não encontrou o melhor valor no tempo máximo de 5 minutos, adotado como critério de parada.

Pode-se também perceber a eficiência dos algoritmos pelo teste de probabilidade empírica. Na Figura 1 está um teste feito com a instância número 8 com $n = 200$ e $m = 80$. No gráfico está representado o tempo gasto para encontrar o melhor valor conhecido e a porcentagem de execuções que conseguiram encontrar esse valor neste tempo. Como nem sempre é possível encontrar esse valor para todas as execuções, nem todas a linhas chegam a 100%. Para ILS1 e ILS2, há só um ponto marcado no gráfico, pois apenas uma das 10 execuções alcançou a meta.

7. CONCLUSÃO

Pode-se concluir com este trabalho que as EE são bons métodos para o PDM, quando comparados com a média dos resultados de métodos conhecidos na literatura como os melhores para o problema. Como uma EE(1+1) com uma busca local tem grande semelhança com o ILS, uma EE(μ,λ) com certeza tem vantagens em relação ao ILS como tem em relação à EE(1+1).

Como descrito nos resultados, é importante notar que as instâncias sem valores de diversidade $d(i, j)$ em função de n podem levar a conclusões precipitadas sobre a qualidade de algoritmos que rapidamente obtêm soluções com qualidade próxima à das melhores soluções. Isso se deve ao fato de que os valores $d(i, j)$ se tornam insignificantes em relação ao problema quando n é muito grande. Neste caso, a qualidade de soluções parecidas com a qualidade do ótimo po-

Table 3: Tempo médio gasto em minutos para encontrar o melhor valor conhecido

Instância	n	m	ILS1	ILS2	ILS3	ILS4	EE
1	100	10	0.27083	0.098051	0.085096	0.050836	0.033397
2	100	20	0.84256	0.22738	0.11232	0.20454	0.04855
3	100	30	0.014695	0.32449	0.018727	0.48963	0.056129
4	100	40	1.9927	0.5105	2.0327	0.21328	0.072837
5	200	20	2.6414	3.3715	2.4588	2.7101	0.42262
6	200	40	5+	4.6327	4.5092	4.6292	1.4873
7	200	60	5+	4.5872	2.8429	4.4913	1.268
8	200	80	5+	4.7111	3.3061	4.1215	0.46233
9	300	30	5+	5+	5+	5+	3.8665
10	300	60	5+	5+	5+	5+	2.7979
11	300	90	5+	5+	5+	5+	2.5811
12	300	120	5+	5+	4.7598	4.604	4.5074
13	400	40	5+	5+	5+	5+	3.4706
14	400	80	5+	5+	5+	5+	4.8054
15	400	120	5+	5+	5+	5+	5+
16	400	160	5+	5+	5+	5+	5+
17	500	50	5+	5+	5+	5+	4.7504
18	500	100	5+	5+	4.6572	5+	4.09
19	500	150	5+	5+	5+	5+	4.6218
20	500	200	5+	5+	5+	5+	5+

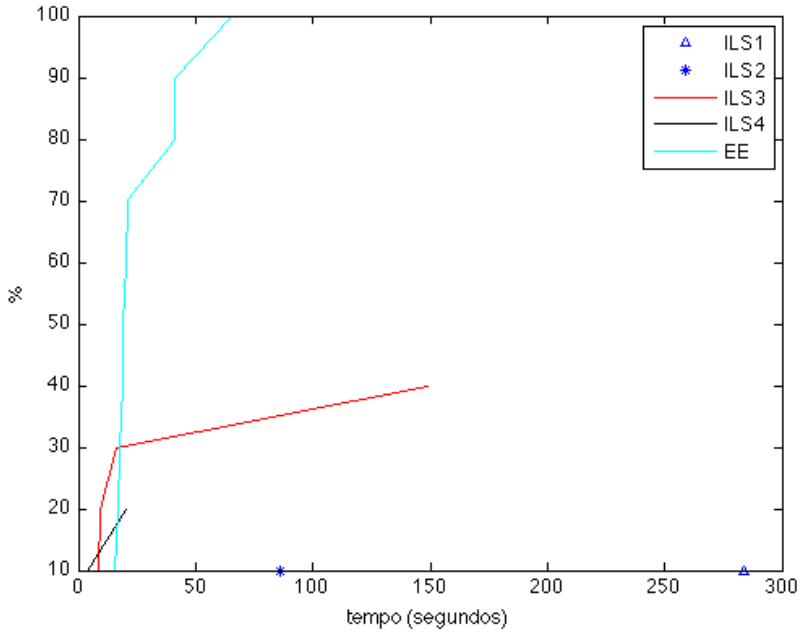

Figure 1: Teste de probabilidade empírica da instância nº 8

dem fazer acreditar que se está próximo de alcançá-lo, o que normalmente não é o caso, devido ao tamanho do espaço de busca que é sempre $n!/m!(n-m)!$. Assim, com este trabalho pode-se mostrar a eficiência das EE também para estes casos.

Agradecimentos

Os autores agradecem à FAPEMIG, CNPq e à CAPES pelo apoio ao desenvolvimento deste trabalho.

REFERENCES

Agrafiotis, D. K. *Journal Chem. Inf. Comput. Sci.* Stochastic algorithms for maximizing molecular diversity, 37, 841-851, 1997

Cutler, M. e Klastorin, T. *Operations Research Letters*. Computacional aspects of the maximum diversity problem, 19, 175-181, 1997.

Duarte, A. E Martí, R. The max diversity problem, Relatório Técnico, University of Washington, Department of Management Science, 1997.

Duarte, I. Silva, G. Costa, T. *Anais do XL Simpósio Brasileiro de Pesquisa Operacional – XL SBPO. João Pessoa, PB*. Algoritmos Heurísticos para o Problema da Diversidade Máxima, 1126-1137, 2008.

Ghosh, J.B. *Operations Research Letters*. Computacional aspects of the maximum diversity problem, 19, 175-181, 1996.

Glover, F., Kuo, C.-C. e Dhir, K. *Journal of Information & Optimization Science,*. Heuristic algorithms for the maximum diversity problem, 109, 132, 1998.

Kuo, C.C., Glover, F. e Dhir, K. *Decision Science*. Analyzing and modeling the maximum diversity problem by zero-one programming, 24, 1171-1185, 1993.

Lourenço, H. R., Martin, O., Stützle,T. *Handbook of Metaheuristics* Iterated Local Search, 57, 321-353, 2003

Martí, R. Disponível em: *http://www.uv.es/rmarti/*. Acesso em: 24 de abril de 2008. Departamento de Estadística e Investigación Operativa de la Universitat de València.

Martí, R., Galego, M., Duarte, A. *European Journal of Operational Research* A branch and bound algorithm for the maximum diversity problem, 200, 1, 36-44, 2010.

Resende, M.G.C. and Martí, R. and Gallego, M. and Duarte, A. *Computers & Operations Research*, Elsevier GRASP and path relinking for the max-min diversity problem, 37, 3, 498-508, 2010.

Resende, M.G.C., Ribeiro, C.C., Glover, F., Martı, R. *Handbook of Metaheuristics, 2nd edn. Springer Science+ Business Media* Scatter search and path-relinking: Fundamentals, advances, and applications, 2010.

Silva, G. C., Ochi, L. S., e Martins, S. L. *Lecture Notes on Computer Science*. Experimental Comparison of Greedy Randomized Adaptive Search Procedures for the Maximum Diversity Problem, 3059, 498-512, 2004.

Silva, G. C., Andrade, M., Ochi, L. S., Martins, S. L e Plastino, A. *Journal of Heuristics*. New heuristics for the maximum diversity problem, 13, 315-336, 2007.

Weitz, R. e Lakshminarayanan, S. *Journal of the Operational Research Society*. An empirical comparison of heuristic methods for creating maximally diverse group, 49, 635-646, 1998.