

OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS COM EXTINÇÃO EM MASSA: UMA ABORDAGEM APLICADA AO PROBLEMA MULTIDIMENSIONAL DA MOCHILA

FERNANDA HEMBECKER, HEITOR S. LOPES, DANIEL ROSSATO

Universidade Tecnológica Federal do Paraná (UTFPR)

Av. Sete de Setembro, 3165, 80.230-901. Curitiba - PR

E-mails: fernanda@denes.com.br, hslopes@pesquisador.cnpq.br, rossato.daniel@gmail.com

Abstract— The multidimensional 0/1 knapsack problem is a classical problem of discrete optimization, and there are several approaches for solving the different variations of such problem, including mathematical programming and heuristic methods. This paper presents the application of a variant of Particle Swarm Optimization (PSO) using mass extinction. PSO, by itself, have presented satisfactory results and the use of mass extinction at given moments of the evolution lead to further accuracy of results. Selected instances of ORLib were used for testing the PSO approach and final results obtained are very close to the optimal known values. Overall, results strongly suggest that PSO is useful and efficient for this class of combinatorial problems.

Keywords— particle swarm optimization, multiple knapsack problem, mass extinction

Resumo— O problema multidimensional da mochila é um problema clássico de otimização. Atualmente, há vários mecanismos para a sua resolução, incluindo diferentes métodos matemáticos discretos e busca estocástica. Este artigo apresenta a aplicação da técnica de Otimização por Enxame de Partículas, com momentos de extinção em massa das partículas, para a resolução do problema da mochila. A técnica de enxame, por si só, apresentou resultados bastante satisfatórios, e provou-se ainda que a extinção das partículas em determinados momentos da busca aprimora estes resultados. Os dados de testes foram selecionados da ORLib e, para as instâncias testadas, os resultados obtidos chegaram muito próximo das soluções ótimas conhecidas. A análise dos resultados sugere o potencial da técnica de otimização por enxame para esta classe de problemas combinatoriais.

Palavras-chave— otimização por enxame de partículas, problema multidimensional da mochila, extinção em massa

1 Introdução

Problemas de otimização são encontrados em diversas aplicações práticas na indústria, em logística e em transportes. Tais problemas são caracterizados como situações onde se deve buscar a melhor configuração ou o melhor conjunto de parâmetros que satisfaçam os critérios de um objetivo específico. Estes problemas vêm sendo estudados amplamente com o apoio da matemática e da computação.

O Problema da Mochila é um problema clássico de otimização cuja descrição breve refere-se a um conjunto de objetos (com respectivos valores de peso e de satisfação) que devem ser colocados em uma recipiente (mochila), obedecendo ao seu limite de capacidade e, ao mesmo tempo, alcançando o maior valor de satisfação possível. Existem, também, algumas variantes desta descrição genérica [11]:

- problema da mochila simples: apenas uma mochila deve conter o melhor conjunto de objetos;
- problema multidimensional da mochila: mais de uma mochila é utilizada;
- problema da mochila de múltipla escolha: os objetos são separados em subconjuntos e no máximo um objeto pode ser selecionado;
- problema da mochila limitado: há uma oferta limitada de objetos a serem selecionados.

Este artigo trata o Problema Multidimensional da Mochila – PMM (*Multidimensional Knapsack Problem – MKP*), possivelmente, a versão mais conhecida do problema [2]. Na literatura, este problema também é intitulado como *Multiconstraint Knapsack Problem*, *Multi-Knapsack Problem*, *Multiple Knapsack Problem* ou *0/1 Multidimensional Knapsack Problem* [9].

O PMM se caracteriza como um problema NP-completo. Isto é, não existe algoritmo que possa resolvê-lo em tempo polinomial. A motivação deste trabalho é buscar um algoritmo heurístico que resolva este problema e que seja efetivo em termos de tempo de processamento e de qualidade de solução. Em [6], foi mostrado que a utilização da técnica de Otimização por Enxame de Partículas – OEP (*Particle Swarm Optimization – PSO*) é eficiente na resolução do problema da mochila. Os resultados obtidos foram bastante satisfatórios, chegando a soluções muito próximas dos ótimos conhecidos. Neste trabalho busca-se aprimorar o algoritmo OEP de forma a melhorar a qualidade das respostas.

Na OEP, partículas percorrem um determinado hiperespaço de busca objetivando encontrar uma combinação de valores que caracterize uma boa solução para o problema tratado. Entretanto, estas partículas podem se agrupar em regiões que representam pontos de máximos locais e deixar de vasculhar o espaço de busca com a abrangência necessária. Para resolver esta questão, propõe-se utilizar a técnica de extinção em massa do enxame ao algoritmo OEP aplicado ao PMM. Acredita-se que desta forma, as

partículas aglomeradas em máximos locais possam explorar outras regiões do espaço de busca e, desta forma, encontrem soluções melhores que as encontradas até então.

2 Otimização por Enxame de Partículas

O algoritmo OEP foi criado por Kennedy e Eberhart [7], sendo que a otimização por enxame de partículas foi inspirada no comportamento de agentes sociais encontrados na natureza, como pode ser observado na revoada de pássaros, no movimento de enxames e de cardumes. Nestes casos, cada indivíduo representa uma entidade única e todos se movimentam harmonicamente no espaço através da influência mútua de cada indivíduo e do grupo como um todo.

Na técnica computacional OEP, agentes ou partículas se movimentam em um espaço de busca multidimensional que representa o universo de soluções para um determinado problema [8]. A movimentação de cada partícula é definida pela memória da própria partícula e também pela influência de todo o grupo (ou de partículas vizinhas).

As partículas são estruturas bastante simples, possuindo capacidade limitada de armazenamento: apenas o seu valor de adequação atual (*fitness*), sua velocidade, sua posição atual e o seu melhor valor de *fitness* encontrado até o momento. Em relação ao *fitness*, o componente cognitivo representa a melhor solução encontrada pela partícula (*best particle solution* - *pbest*); o componente social representa a melhor solução encontrada pelo enxame (*best global solution* - *gbest*) [8]. Em alguns casos se utiliza como componente social a melhor solução local, que representa a melhor solução em uma vizinhança de partículas (*best local solution* - *lbest*). Na maioria dos casos, para considerar a influência de partículas vizinhas, utiliza-se o *gbest* ao invés do *lbest*.

As partículas são inseridas em um processo evolutivo onde, a cada iteração, definem o seu novo posicionamento no espaço de busca. Desta forma, a cada iteração, são definidos novos valores de *pbest* e *gbest* que passam a influenciar o próximo movimento das partículas. Como a OEP se caracteriza como um método de busca heurística baseado em população, é preciso garantir a manutenção da diversidade do enxame durante o processo evolutivo. Esta diversidade é fundamental para garantir que o espaço de busca seja explorado o mais eficientemente possível. Há situações em que o *pbest* representa um ponto no espaço com bom valor de *fitness* e está distante da localização do *gbest*, que representa outra boa resposta. Neste caso, observa-se a diversidade do enxame. Porém, há situações onde o *pbest* e o *gbest* se referem a partículas muito próximas, levando o enxame a ficar estagnado em uma região limitada do espaço de busca.

Em resumo, segundo [5], o algoritmo que representa a versão global de OEP é:

- a) Iniciar uma população de partículas com posições e velocidades aleatórias no espaço n -dimensional que represente as soluções do problema.
- b) Para cada partícula, calcular seu valor de *fitness* em função de suas n dimensões.
- c) Comparar o *fitness* da partícula com o componente cognitivo - *pbest*. Se o valor atual for melhor que o *pbest*, atualizar o valor de *fitness* e a localização do *pbest* com os respectivos valores da partícula atual.
- d) Comparar o *fitness* da partícula atual com o componente social - *gbest*. Se o valor atual for melhor que o *gbest*, atualizar o valor de *fitness* e a localização do *gbest* com os respectivos valores da partícula atual.
- e) Alterar a velocidade e a posição da partícula, levando em consideração sua posição atual, sua velocidade e os componentes cognitivo e social.
- f) Repetir os passos a partir do item (b) até atingir um determinado critério de parada, sendo este um valor de *fitness* esperado ou um número determinado de iterações.

No modelo clássico da OEP, a movimentação da i -ésima partícula é definida pela equação 1, onde a próxima posição no espaço de busca (X_i^{t+1}) é atualizada de acordo com sua posição atual e a sua velocidade v_i^t .

$$X_i^{t+1} = X_i^t + V_i^t \quad (1)$$

De acordo com [5], por questões de simplificação, usa-se a expressão velocidade para representar, na verdade, ΔX_i . A velocidade propriamente dita não é considerada neste caso, sendo, então, o termo V_i^t definido de acordo com a equação 2:

$$V_i^t = c_1 \cdot r_1 \cdot \Delta_{pbest} + c_2 \cdot r_2 \cdot \Delta_{gbest} \quad (2)$$

onde r_1 e r_2 são valores aleatórios no intervalo [0..1]; c_1 e c_2 são os pesos de *pbest* e *gbest*, respectivamente; e Δ_{pbest} e Δ_{gbest} representam as distâncias: entre a posição atual da partícula e *pbest* e entre a posição atual da partícula e *gbest*, respectivamente. Observa-se desta forma que o termo velocidade representa o fator de atualização da posição da partícula, sendo diretamente proporcional à distância entre *pbest* e *gbest*. Desta forma, com o passar das iterações, a partícula será atraída para a região de *pbest* ou de *gbest*.

A equação (2) reflete também os graus de busca local e global da partícula. Valores mais altos de velocidade incitam a exploração global, ao passo que valores menores incitam a exploração local. Idealmente, não se deve determinar valores muito altos ou muito pequenos para a velocidade, uma vez que, no primeiro caso, regiões do espaço de busca podem ser desconsideradas e, no segundo caso, a busca se torna exaustiva. O algoritmo do PSO permite então a defi-

nição de um valor máximo de velocidade (V_{\max}) para controlar esta situação.

Como exemplo, considerando o espaço de busca bi-dimensional, a figura 1 apresenta os elementos que influenciam a movimentação da partícula no espaço de busca.

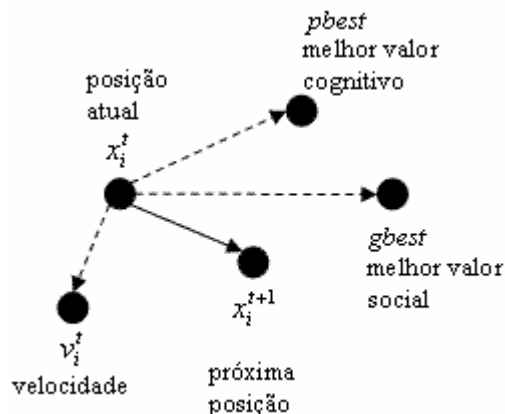


Figura 1. Influências na movimentação das partículas.

Como afirmado anteriormente, cada partícula é considerada um indivíduo que atua em um grupo, sendo esta a interpretação psicológica do algoritmo OEP [5]. Desta forma, segundo a equação (2), a influência do $pbest$ faz com que a partícula tenda a seguir o seu melhor comportamento atingido em iterações passadas. De forma semelhante, a influência de $gbest$ faz com que a partícula considere o sucesso obtido por outra partícula do enxame. Aquela equação exige, ainda, a definição das constantes c_1 e c_2 que atuam diretamente na influência de $pbest$ e $gbest$, respectivamente. Na literatura não há sugestão de valores-padrão para estas constantes, que podem possuir valores simétricos ou assimétricos. Entretanto, pesquisas sugerem que estes valores devem ser definidos de acordo com as características do problema a ser resolvido [3].

3 Metodologia

No PMM, considera-se a existência de n objetos e m mochilas de capacidades c_j ($j=1, \dots, m$); variáveis binárias x_i ($i=1, \dots, n$) que assumem o valor 1, caso o objeto i seja selecionado para constar nas mochilas, ou o valor 0 em caso contrário. Todo objeto possui um valor de satisfação p_i e, para cada mochila, um peso w_{ij} específico. O problema de otimização é então caracterizado pelas equações:

$$\text{maximizar } \sum_{i=1}^n p_i x_i \quad (3)$$

$$\text{sujeito a } \sum_{j=1}^m w_{ij} x_i \leq c_j \quad i = 1, \dots, n \quad (4)$$

$$\text{onde } x_i = 0 \text{ ou } 1 \quad i = 1, \dots, n$$

O objetivo do PMM é colocar nas mochilas a seleção ideal de objetos que apresente o maior grau de satisfação, sem que a capacidade das mesmas seja extrapolada [9]. Considerando a aplicação da OEP, cada partícula deve representar uma possível solução do problema. Desta forma, elas são codificadas como vetores de números binários onde cada elemento representa a seleção (1) ou a ausência (0) do objeto. O espaço n dimensional de busca é definido pela quantidade de objetos disponíveis no problema e considera-se, ainda, que um dado objeto selecionado é contabilizado em todas as mochilas.

De maneira semelhante a algoritmos genéticos [4], eventualmente uma partícula pode representar uma solução inválida para o problema. Isto ocorre porque a otimização é avaliada pela equação (3), mas a restrição, apresentada na equação (4), não é diretamente manipulada pelo algoritmo. Este tipo de partícula é permitida na OEP, uma vez que o processo evolutivo e as influências do $pbest$ e do $gbest$ podem tornar uma partícula inválida em uma partícula válida. Desta forma, a partícula não é descartada, mas seu valor de *fitness* sofre uma penalidade alta de modo que não influencie demasiadamente as outras partículas. Esta penalidade é calculada proporcionalmente em função do valor excedido em relação ao limite das mochilas.

Como em outros sistemas de busca baseados em população, um dos maiores desafios é manter a diversidade desta população. Segundo [17], as formas mais estudadas de se manter a diversidade em algoritmos evolucionários são:

- criar estruturas complexas de população para reduzir a transferência entre genes (ex.: modelo de ilhas);
- utilizar operadores especializados para controlar e assistir os procedimentos de seleção (ex.: *crowding*);
- reintroduzir material genético periodicamente (ex.: modelos de extinção em massa).

Neste trabalho, aplicou-se a extinção em massa do enxame quando detectada a convergência prematura para uma única região no espaço de busca. Com esta extinção (ou “explosão” do enxame), as partículas são reposicionadas aleatoriamente no espaço e continuam com o processo de busca. Neste caso, apenas a posição atual das partículas é alterada, sendo mantidos os valores de $pbest$ e $gbest$ encontrados em iterações anteriores. Desta forma, garante-se que as partículas serão atraídas novamente a boas regiões do espaço de busca, mas, por tomarem novos caminhos, podem descobrir soluções melhores que as anteriores e desviar das regiões de máximo local.

Acredita-se que desta forma seja possível controlar a diversidade do enxame. Quando as partículas se concentram em uma região, $pbest$ e $gbest$ se encontram muito próximos. Neste caso, a busca local se sobressai em relação à busca global e, assim, corre-se o risco de não explorar outras regiões do espaço com possíveis soluções melhores que a solução local.

4 Experimentos Computacionais

Para os experimentos, foram consideradas instâncias do problema multidimensional da mochila disponibilizados na ORLib [1]. Os grupos selecionados foram: Sento [13], Weing [16] e Weish [14], com 2, 8 e 30 problemas, respectivamente. O grau de dificuldade dos mesmos varia de acordo com a quantidade de mochilas e objetos que cada conjunto apresenta. Os conjuntos Sento possuem 30 mochilas e 60 objetos. Os conjuntos Weing possuem todos 2 mochilas e 28 ou 105 objetos; e os conjuntos Weish, por sua vez, possuem todos 5 mochilas e variam de 30 a 90 objetos.

Inicialmente, os testes foram efetuados para todas as instancias do problema da mochila sem que fosse efetuada a extinção do enxame. Neste caso, 500 iterações foram rodadas para cada instância, resultando em 20.000 rodadas.

Posteriormente foram efetuados testes com a extinção em massa do enxame. Para tanto, foi considerada convergência prematura situações onde o *fitness* médio do enxame se manteve constante por um determinado número de iterações. Para definir este número, ou seja, o momento exato da extinção, todas as instâncias foram testadas com extinção a cada 10, 20, 40, 60, 80, 120, 200 iterações sem melhora do *fitness* médio. Neste caso, 140.000 rodadas foram realizadas e observou-se que, na média, os melhores resultados foram obtidos com explosão a cada 20 iterações sem melhora.

Outro ponto considerado nos testes foi a variação de c_1 e c_2 e V_{max} para cada conjunto de problemas. Validando a hipótese de que cada conjunto deve trabalhar com valores específicos destas constantes [3], o caso mais complexo de cada instância foi testado com os seguintes valores: 0.5, 1.0, 1.5 e 2.0 para c_1 e c_2 , e 1.0, 1.5 e 2.0 para V_{max} . Os experimentos indicaram que o melhor grupo de valores para cada instância é:

- Sento: $c_1=2.0$; $c_2=0.5$; $V_{max}=1.5$
- Weing: $c_1=2.0$; $c_2=0.5$; $V_{max}=2.0$
- Weish: $c_1=2.0$; $c_2=0.5$; $V_{max}=2.0$

Uma vez definidos os valores das constantes e do momento exato da extinção, as instâncias foram testadas e os resultados obtidos para a instância Weish são apresentados na tabela 1. A figura 2 apresenta o resultado dos testes efetuados sobre a instância Weing, sendo mostrado o percentual de melhora das soluções obtidas utilizando a extinção em massa (com referência ao mesmo problema sem a extinção). Nesta figura é mostrada a melhora obtida tanto para o melhor resultado obtido, quanto para a média das diversas rodadas.

Tabela 1: Resultados para a instância Weish

conjuntos weish	objetos	ótimo conhecido	melhor sem explosão	melhor com explosão	diferença absoluta sem explosão	diferença absoluta com explosão	melhora na média com explosão
1	30	4554	4554	4554	0,00%	0,00%	0,20%
2	30	4536	4536	4536	0,00%	0,00%	0,41%
3	30	4115	4115	4115	0,00%	0,00%	0,45%
4	30	4561	4561	4561	0,00%	0,00%	0,10%
5	30	4514	4514	4514	0,00%	0,00%	0,03%
6	40	5557	5557	5557	0,00%	0,00%	0,38%
7	40	5567	5567	5567	0,00%	0,00%	0,62%
8	40	5605	5605	5605	0,00%	0,00%	0,56%
9	40	5246	5246	5246	0,00%	0,00%	0,28%
10	50	6339	6339	6339	0,00%	0,00%	0,34%
11	50	5643	5643	5643	0,00%	0,00%	0,72%
12	50	6339	6339	6339	0,00%	0,00%	0,71%
13	50	6159	6159	6159	0,00%	0,00%	0,56%
14	60	6954	6954	6954	0,00%	0,00%	0,88%
15	60	7486	7486	7486	0,00%	0,00%	1,07%
16	60	7289	7289	7289	0,00%	0,00%	0,67%
17	60	8633	8633	8633	0,00%	0,00%	1,00%
18	70	9580	9569	9580	11,48%	0,00%	1,94%
19	70	7698	7698	7698	0,00%	0,00%	1,43%
20	70	9450	9445	9450	5,29%	0,00%	1,17%
21	70	9074	9074	9074	0,00%	0,00%	1,34%
22	80	8947	8929	8929	20,12%	20,12%	1,45%
23	80	8344	8344	8344	0,00%	0,00%	2,57%
24	80	10220	10169	10215	49,90%	4,89%	1,74%
25	80	9939	9919	9939	20,12%	0,00%	1,57%
26	90	9584	9532	9584	54,26%	0,00%	3,06%
27	90	9819	9816	9819	3,06%	0,00%	2,11%
28	90	9492	9469	9492	24,23%	0,00%	4,54%
29	90	9410	9337	9410	77,58%	0,00%	4,11%
30	90	11191	11102	11191	79,53%	0,00%	4,33%

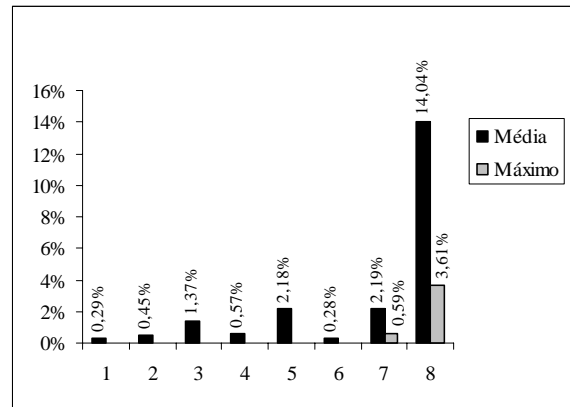


Figura 2. Melhoras obtidas com explosão sobre a instância Weing.

5 Conclusões e Trabalhos Futuros

De acordo com a tabela 1, o desempenho médio do PSO sem extinção, nos 30 casos da instância Weish, foi de 11,83% abaixo do ótimo conhecido. Além disto, em 66,67% dos casos, atingiu-se o ótimo. Por outro lado, para o PSO com extinção em massa, o desempenho foi de 0,83% do ótimo conhecido, sendo que em 93,33% dos casos o ótimo foi encontrado.

O valor médio do *fitness* das partículas com a aplicação da extinção teve uma melhoria de 1,35%. Mesmo sendo um valor relativamente baixo, observou-se que a melhoria do desempenho (diferença absoluta com explosão) foi significativa, o que indica que a manutenção da diversidade do enxame efetivamente melhora os resultados. A melhoria do *fitness* médio também se observa nos conjuntos Weing, como mostrado na figura 2. Para o conjunto Weing8, a melhoria no *fitness* médio em 14.04% ocasionou a melhoria em 3.61% no máximo encontrado.

No trabalho proposto por Beasley [2] utilizou-se algoritmos genéticos para resolver o problema da mochila. Para os conjuntos Sento, Weing e Weish, foram encontrados os valores ótimos, mas para isso, foi necessário realizar iterações até que 10.000 indivíduos não duplicados fossem gerados. Neste caso, a resposta ótima é encontrada, mas o custo computacional é bastante elevado.

Nos testes efetuados, constatou-se que a definição de valores específicos de c_1 e c_2 e V_{max} para cada conjunto de problemas afeta os resultados. Ou seja, a OEP é sensível aos seus parâmetros de controle e este fato altera o modo como o espaço de busca é percorrido [15]. Em trabalhos futuros pretende-se estudar a variação dinâmica da influência dos componentes cognitivo e social em diferentes momentos da busca. Ou seja, com a utilização de estratégias auto-adaptativas [12], pode-se refinar o tratamento dos parâmetros a fim de obter um controle mais eficiente da movimentação das partículas. Em [10], apresenta-se uma hibridização do método OEP com busca local, levando à obtenção de resultados melhores do que os obtidos com a OEP pura. Esta também será uma outra linha de pesquisa futura.

De forma geral, pode-se dizer que a utilização de heurísticas independentes do problema – como o caso da OEP, garantem a eficiência na varredura do espaço de busca para problemas complexos. Mesmo sensíveis a alguns parâmetros de configuração, podem ser vistas como técnicas promissoras na resolução de problemas combinatoriais.

Agradecimento

O autor H.S. Lopes agradece o CNPq pelo apoio através da bolsa PQ, processo 305720-2004-0.

Referências Bibliográficas

1. Beasley, J.E. (2005) *ORLib – Operations Research Library*. [<http://people.brunel.ac.uk/~mastjjb/jeb/orlib/files/mknap2.txt>].
2. Beasley, J.E. e Chu, P.C. (1998) Genetic algorithm for the multidimensional knapsack problem. *Journal of Heuristics* **4**:63–86.
3. Clerc, M. (1999) The swarm and the queen: towards a deterministic and adaptive particle

- swarm optimization. *Proc. IEEE Congress on Evolutionary Computation*, 3, pp.1951–1957.
4. Eberhart, R.C. e Shi, Y. (1998) Comparison between genetic algorithms and particle swarm optimization. *Proc. 7th Annual Conference on Evolutionary Programming*, pp. 611–616.
5. Eberhart, R.C. e Shi, Y. (2001) Particle swarm optimization: developments, applications and resources. *Proc. Congress on Evolutionary Computation*, 1, pp. 81–86.
6. Hembeker, F., Lopes, H.S., e Godoy Jr., W. (2007) Particle swarm optimization for the multidimensional knapsack problem. *Proc. 8th Int. Conf. Adaptive and Natural Computing Algorithms*, part I, LNCS v. 4431, pp. 358–365.
7. Hoff, A., Løkketangen, A. e Mittet, I. (1996) Genetic algorithm for 0/1 multidimensional knapsack problems. *Proc. Norsk Informatikk Konferanse*, [s.p.].
8. Kennedy, J. e Eberhart, R.C. (2001) *Swarm Intelligence*. Morgan Kaufmann, San Francisco.
9. Khuri, S., Bäck, T. e Heitkoetter, J. (1994) The zero/one multiple knapsack problem and genetic algorithm. *Proc. ACM Symposium on Applied Computing*, Phoenix, USA, pp. 188–193.
10. Lopes, H.S. e Coelho, L.S. (2005) Particle swarm optimization with fast local search for the blind travelling salesman problem. *Proc. 5th Hybrid Intelligent Systems Conference*, Rio de Janeiro, Brazil, pp. 245–250.
11. Martelo, S. e Toth, P. (1990) *Knapsack Problems - Algorithms and Computer Implementations*. John Wiley & Sons, New York.
12. Maruo, M.H., Lopes, H.S. e Delgado, M.R.B.S. (2005) Self-adapting evolutionary parameters: encoding aspects for combinatorial optimization problems. *Proc. Evolutionary Computation for Combinatorial Problems*, LNCS, v. 3448, pp. 154–165.
13. Senyu, S. e Toyoda, Y. (1967) An approach to linear programming with 0-1 variables. *Management Science*, 15:B196–B207.
14. Shih, W. (1979) A branch and bound method for the multiconstraint zero-one knapsack problem. *Journal of the Operational Research Society*, **30**:369–378.
15. Trelea, I.C. (2003) The particle swarm optimization algorithm: convergence analysis and parameter selection. *Information Processing Letters*, **85**(6):317–325.
16. Weingartner, H.M. e Ness, D.N. (1967) Methods for the solution of multidimensional 0/1 knapsack problems. *Operations Research*, **15**:83–103
17. Ursem, R.U., (2002) Diversity-guided evolutionary algorithms. *Proc. of Parallel Problem Solving from Nature*, LNCS, v. 2439, pp. 462–471