

Meta-heurística Híbrida de Sistema de Colônia de Formigas e Algoritmo Genético para o Problema do Caixeiro Viajante

M.B. CARVALHO¹, A. YAMAKAMI², Departamento de Telemática, Faculdade de Engenharia Elétrica e de Computação, UNICAMP, 13083-852 Campinas, SP, Brasil.

Resumo. Apresentamos neste artigo, uma nova Meta-heurística Híbrida de Sistema de Colônia de Formigas (ACS) e Algoritmos Genéticos (AG) para resolver o Problema do Caixeiro Viajante (PCV). A resolução do Problema do Caixeiro Viajante é complexa, pois envolve uma busca em um enorme espaço de soluções que cresce conforme aumenta o número de nós do grafo, tornando inviável a utilização de métodos exatos. O Algoritmo Híbrido ACS+AG-PCV é proposto visando obter bons resultados, de maneira a contornar a questão da complexidade do Problema do Caixeiro Viajante.

Palavras-chave. Ant Colony System. Algoritmo Genético. Problema do Caixeiro Viajante.

1. Introdução

Modelos baseados em sistemas naturais são bem sucedidos para problemas de otimização combinatorial *NP*-difícil [7]. A Meta-heurística de Otimização por Colônia de Formigas (ACO) introduzida no início da década de 90 por Dorigo [4], é uma estrutura genérica para Algoritmos de Otimização baseada em Colônia de Formigas. Resultados promissores de adaptações da Meta-heurística de Otimização por Colônia de Formigas (ACO) vêm ocorrendo em diversos problemas combinatoriais, tais como, Caixeiro Viajante [6], roteamento de veículos [3], entre outros. Algoritmos de Otimização por Colônia de Formigas, tais como o Sistema de Colônia de Formigas (ACS)[5], são usados com sucesso em diversas instâncias do PCV. O PCV consiste na determinação de uma rota que inicia em uma cidade i , passa por cada cidade j do conjunto apenas uma vez, e retorna à cidade inicial da rota percorrendo uma distância total mínima. Esta rota é denominada ciclo Hamiltoniano de custo mínimo. A dificuldade em resolver esse tipo de problema está no número elevado de soluções existentes. Em muitos casos, é praticamente impossível garantir a solução ótima global do problema e, somente com a utilização de heurísticas e meta-heurísticas é possível obter soluções aproximadas [11]. Entre as possíveis

¹marciab@dt.fee.unicamp.br

²akebo@dt.fee.unicamp.br

heurísticas, estão aquelas que imitam processos naturais em busca de equilíbrio, em geral otimizado, como ACS, AG, etc.

Neste artigo propomos um Algoritmo Híbrido ACS+AG-PCV, que trabalha com as Meta-heurísticas ACS [2] e AG [8] conjuntamente, para resolver o PCV. O método proposto consiste no processo de cooperação entre os algoritmos, e esse processo ocorre no sentido em que ACS exporta soluções potenciais para a população de AG sempre que não é verificada nenhuma melhoria nas soluções encontradas pelas formigas, e AG reforça alternativas de buscas para as formigas artificiais encontrando melhores soluções do que as existentes. Esse reforço é dado através da aplicação do feromônio no menor caminho encontrado pelo AG.

2. Literatura do PCV

Atualmente existe um número muito grande de problemas que podem ser modelados como um PCV. Com isso, abordagens exatas e de aproximações, como as heurísticas e meta-heurísticas, têm sido buscadas para a resolução de instâncias de grande porte desse problema, tanto para o caso do PCV simétrico quanto para o assimétrico.

Técnicas que são freqüentemente utilizadas para resolver métodos exatos do PCV são *branch-and-bound* e *branch-and-cut*.

Entre os métodos heurísticos para a resolução do PCV podemos citar alguns deles:

Dorigo e Gambardella [5] propuseram a heurística de Otimização por Colônia de Formigas, que é inspirada na observação das colônias de formigas reais e baseia-se na capacidade das formigas encontrarem um caminho mínimo entre a colônia e a fonte de alimento. Neste artigo, são mostradas comparações entre a heurística de ACO e os métodos de otimização global, como: Simulated Annealing, Rede elástica e Algoritmos Genéticos, sendo que os resultados da heurística de Colônia de Formigas foram melhores que os três métodos para as matrizes de custos consideradas. Além disso, a heurística proposta é aplicada com a técnica de melhoria *3-opt* [9], reduzindo os erros médios que se encontravam nos grafos considerados.

O trabalho de Pilat e White [12] mostra a aplicação de um método híbrido de Algoritmos Genéticos aplicado ao método de Colônias de Formigas para o PCV simétrico. Os Algoritmos Genéticos são utilizados para determinar quais formigas artificiais servem para melhorar as soluções para o problema.

No trabalho de Merz e Freisleber [1] é apresentada a aplicação de Algoritmos Genéticos para o PCV simétrico, com aplicação da técnica de Busca local *2-opt* [9] para a rota encontrada, com o objetivo de encontrar melhores rotas para as próximas gerações. Os resultados encontrados no banco de dados do *TSPLIB* [13] mostram que esta técnica é capaz de obter soluções na maioria dos casos citados.

Affenzeller e Wagner [10] mostram uma técnica para determinar boas soluções para o PCV com o uso de AG através das comparações entre o *fitness* de novas soluções com o *fitness* da população e dos *fitness* dos novos indivíduos e de seus pais. Os resultados para alguns problemas do *TSPLIB* [13] são melhores do que os resultados encontrados com a aplicação de AG da forma tradicional.

O trabalho de Sari, Sherali e Bhootra [14] mostra uma nova formulação para

o PCV assimétrico, criando restrições para a criação de sub-rotas. Os resultados para alguns problemas do *TSPLIB* [13] mostram que esta nova formulação permite a obtenção de soluções ótimas em todos os casos testados.

Do nosso conhecimento, ao contrário dos algoritmos de Otimização por Colônia de Formigas e AG para o PCV que já foram bastantes explorados individualmente nesse problema, praticamente não existem nenhum trabalho utilizando as Meta-heurísticas de ACS e AG conjuntamente. Por isso apresentamos um novo Algoritmo Meta-heurístico Híbrido reunindo os algoritmos ACS e AG para resolver o PCV.

3. Descrição do Algoritmo ACS+AG-PCV

Os principais objetivos do Algoritmo Híbrido proposto neste artigo são melhorar a capacidade de busca do algoritmo, a velocidade de convergência e a qualidade das soluções. Na abordagem ACS+AG-PCV há um processo de cooperação entre os algoritmos ACS e o AG. Esse processo se dá através de uma migração das soluções encontradas pelos algoritmos sempre que algum deles não consegue sair de um ponto de mínimo local. A migração fornece uma capacidade adicional de intensificação para ambos os algoritmos.

A estratégia de migração implementada é a seguinte: sempre que o algoritmo de formigas, o ACS, não consegue sair de um ponto ótimo local, ele exporta uma quantidade pré-determinada de soluções encontradas até o momento (melhores soluções encontradas pelas formigas), para formar a população inicial do algoritmo genético. Neste momento as iterações do ACS são interrompidas e o AG tem início para encontrar uma melhor solução que a do ACS. Uma vez que essa solução é encontrada pelo AG, ele exporta essa solução para o algoritmo ACS adicionar feromônio nesse novo caminho, reforçando as alternativas de buscas para as formigas artificiais. Esse processo se repete até que a melhor solução conhecida (disponível no *TSPLIB* [13]) seja encontrada ou se atinja o número máximo de iterações dos algoritmos.

Como o Algoritmo Híbrido proposto neste artigo combina as Meta-heurísticas de Colônia de Formigas e o AG, iremos detalhar o funcionamento desses algoritmos: inicialmente iremos apresentar o algoritmo de formigas e depois apresentaremos a estrutura do algoritmo genético para o problema do PCV.

O algoritmo de formigas [2] implementado para o PCV pode ser descrito da seguinte forma: para cada arco (i, j) do grafo, é definida uma variável τ_{ij} , conhecida como trilha artificial de feromônio. Inicialmente este τ_{ij} é igual para todos os arcos da rede. Cada formiga k constrói uma solução a partir de um dos nós do grafo da seguinte maneira:

1. uma formiga posicionada na cidade i escolhe a cidade j para visitar, dentre as que ainda não foram visitadas, de acordo com a regra abaixo

$$j = \operatorname{argmax}_{u \in J_i^k} \{[\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta\}, \quad \text{se } q \leq q_0.$$

2. ou ainda se $q > q_0$, a probabilidade da formiga que esta na cidade i , escolher a cidade j para visitar é diretamente proporcional à quantidade τ_{ij} de feromônio

na aresta (i, j) e inversamente proporcional ao tamanho da aresta;

$$p_{ij}^k = \frac{[\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{l \in J_i^k} [\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta}, \quad j \in J_i^k, \quad \text{se } q > q_0,$$

onde

p_{ij}^k : é a probabilidade da formiga k , que se encontra na cidade i , escolher o nó j como próximo nó a ser visitado;

τ_{ij} : quantidade de feromônio existente no arco (i, j) . Inicialmente, adota-se um mesmo valor τ_0 para todos os arcos da rede;

η_{ij} : função heurística que representa a atratividade do arco (i, j) . No caso do PCV, adota-se o inverso $1/c_{ij}$ do valor da distância entre os nós i e j ;

J_i^k : conjunto de pontos ainda não visitados pela formiga k , que se encontra atualmente no ponto i ;

α : é um parâmetro que pondera a importância relativa da trilha de feromônio, τ_{ij} na decisão de movimentação da formiga.

β : valor heurísticamente escolhido, que pondera a influência relativa da distância η_{ij} entre os nós i e j no processo de decisão.

q : variável aleatória distribuída uniformemente entre $[0, 1]$.

q_0 : é um parâmetro ($0 < q_0 < 1$).

3. a formiga tem a capacidade de se lembrar de cada cidade já visitada;

4. cada formiga, após completar um ciclo, atualizará o feromônio nas arestas (i, j) ;

5. a atualização do feromônio é feita local e globalmente. Essa atualização local pretende evitar uma aresta muito forte que está sendo escolhida por todas as formigas, e também é motivada pela evaporação do feromônio:

$$\tau_{ij} = (1 - \rho) \cdot \tau_{ij} + \rho \cdot \tau_0, \quad \rho \in [0, 1],$$

onde ρ é um parâmetro que determina a velocidade da evaporação do feromônio.

6. a atualização global pretende recompensar as arestas que pertencem a rotas mais curtas:

$$\tau_{ij} = (1 - \rho) \cdot \tau_{ij} + \rho \cdot \Delta \tau_{ij},$$

$$\Delta \tau_{ij} = \begin{cases} \frac{1}{L^{max}}, & \text{se } (i, j) \in S^* \\ 0, & \text{caso contrário} \end{cases}$$

onde

$\Delta \tau_{ij}$ é a quantidade de feromônio que será depositado na aresta (i, j) ;

L^{max} é o custo total da melhor solução S^* .

O AG implementado no Algoritmo Híbrido é apresentado abaixo:

Representação: cada cromossomo representa um *circuito*, ou seja, uma solução para o problema, e seus genes representam os nós associados aos custos das arestas que ligam esses nós. No algoritmo genético desenvolvido para o algoritmo híbrido, cada cromossomo $C_i = (c_{i1}, c_{i2}, c_{i3}, \dots, c_{in})$ representa um circuito (*tour* do caixeiro viajante) e n representa o número de cidades (Figura 1).

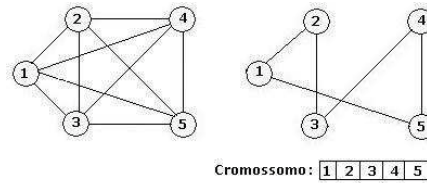


Figura 1: Exemplo de um cromossomo representando um *tour* do PCV

Inicialização: a população inicial do AG é dada pelos caminhos percorridos pelas formigas no algoritmo ACS quando este não obtém melhorias.

Crossover: o *crossover* é feito escolhendo-se dois indivíduos pais #P1 e #P2 da população para gerar dois indivíduos filhos #F1 e #F2. Primeiro sorteia aleatoriamente um trecho de cada um dos pais; depois copia o trecho do pai #P1 para o filho #F2 e o do pai #P2 para o filho #F1; o restante do cromossomo do filho #F1 e #F2 é preenchido com as informações do outro pai #P2 e #P1 respectivamente, na mesma seqüência para evitar valores repetidos nos genes (Figura 2).

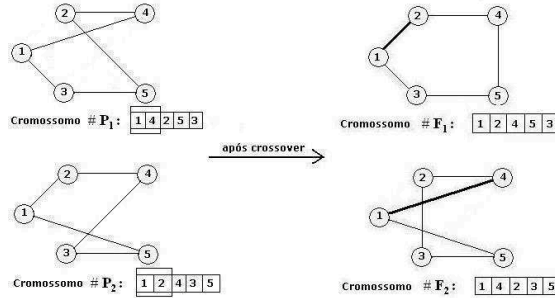


Figura 2: Exemplo de *crossover* entre dois pais #P1 e #P2

Mutação: o operador de mutação utilizado foi a mutação inversiva, onde escolhe aleatoriamente um indivíduo da população, seleciona dois pontos aleatórios do cromossomo e seus genes são permutados (Figura 3).

Função de Avaliação (*Fitness*): a função de avaliação associa a cada indivíduo da população uma medida de aptidão. Ela deve ser escolhida de forma a medir o desempenho de cada indivíduo como solução do problema. Neste algoritmo adotou-se um *fitness* igual ao custo do circuito, ou seja o custo de cada viagem.

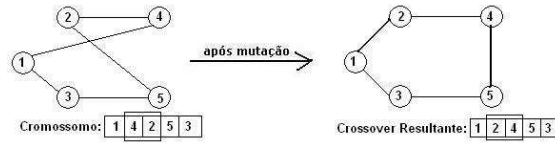


Figura 3: Exemplo de mutação em um *tour* do PCV

$$fitness(C_i) = \sum_{i,j=1}^n c_{ij}$$

Diversidade: a chance de obter boas soluções aumenta em uma população de indivíduos mais diversificados. Para analisar o comportamento da diversidade da população, foi definida a seguinte medida: *Diversidade Populacional* é a razão entre o número de indivíduos únicos (aqueles que não se repetem na população) e o tamanho da população. Desta maneira é possível identificar quando o algoritmo esta perdendo diversidade.

Variabilidade Genética: segundo Michalewicz [11] a variabilidade de uma população cai conforme o *fitness* aumenta. Isso ocorre porque a chance dos indivíduos gerarem descendentes é maior para aqueles que possuem *fitnesses* maiores. Com isso, o algoritmo utiliza os seguintes mecanismos para injetar variabilidade na população:

Seleção Bi-Classista: a cada geração é preservada $P\%$ dos melhores indivíduos e $(100 - P\%)$ dos piores indivíduos da população.

Imigração: em cada geração é injetada uma porcentagem de novos indivíduos, gerados de forma aleatória.

4. Resultados Computacionais

Nesta seção são apresentados os resultados obtidos com o Algoritmo Híbrido para instâncias do PCV simétrico. Além deste algoritmo, implementamos também um AG clássico e o ACS [2], a fim de realizar uma comparação entre os algoritmos.

Para avaliar o desempenho do algoritmo proposto, foram utilizados 6 grafos completos com distâncias simétricas retirados da base de dados *TSPLIB* [13]. O *TSPLIB* [13] é um repertório público de instâncias para o PCV na Internet onde, além das instâncias, consta também o ótimo global para algumas delas.

Para cada grafo foram realizados 10 experimentos e os resultados obtidos com o algoritmo proposto foram comparados com os dos algoritmos AG e o ACS (Tabela 1). Os parâmetros utilizados nos experimentos do Algoritmo Híbrido foram:

- ACS+AG-PCV: número máximo de iteração = 500, número de formigas = 20, $\beta = 2$, $\alpha = 1$, $\rho = 0,1$, $q_0 = 0,9$. Critério de parada ACS: encontrar a solução ótima ou número máximo de iterações; número máximo de gerações = 600, tamanho

da população = 60, $T_c = 60\%$ (taxa de *crossover*), $T_m = 3\%$ (taxa de *mutação*), $P = 60\%$. Critério de parada AG: encontrar melhor solução que o ACS ou número máximo de gerações.

Para os 10 experimentos feitos com cada problema, a Tabela 1 apresenta o custo total da melhor solução e da pior solução encontrada, bem como a média dos custos das soluções. Os grafos das instâncias utilizadas neste trabalho representam conexões existentes entre cidades, variando em um número de cidades no intervalo [51,226]. Desta forma as instâncias testadas possuem em torno de $3,04 \times 10^{64}$ a $1,25 \times 10^{433}$ soluções possíveis.

Na Tabela 1 estão os resultados obtidos com os algoritmos ACS+AG-PCV, ACS e o AG. Em negrito são indicados os resultados que atingiram a melhor solução conhecida (solução ótima) durante todos os experimentos.

Tabela 1: desempenho dos algoritmos ACS+AG-PCV, ACS e AG.

Problema	Algoritmo	Melhor solução encontrada	Média das soluções	Pior solução encontrada	Tempo (s)
eil51 (51 nós)	ACS+AG-PCV	426	426,4	427	25
	ACS	426	426,8	430	
	AG	429	435	440	
brazil58 (58 nós)	ACS+AG-PCV	25395	25398	25405	87
	ACS	25395	25517	25643	
	AG	25395	25578	25966	
eil76 (76 nós)	ACS+AG-PCV	538	540	543	114
	ACS	543	547,3	549	
	AG	551	564,7	573	
kroA100 (100 nós)	ACS+AG-PCV	21282	21291	21307	270
	ACS	21320	21471	21597	
	AG	21395	21679	21970	
bier127 (127 nós)	ACS+AG-PCV	118282	118958	119458	1140
	ACS	119206	120418,8	121332	
	AG	120884	124816,7	126240	
pr226 (226 nós)	ACS+AG-PCV	80369	80505,9	80644	1260
	ACS	81243	82997	82490	
	AG	81904	83680,5	84782	

A Figura 4(a) mostra o gráfico de convergência do Algoritmo Híbrido ACS+AG-PCV para o grafo bier127. Nele podemos observar a contribuição do AG na solução do Algoritmo de Formigas ajudando a sair de mínimos locais. A escolha das Figuras 4(a) e (b) abaixo, é apenas para ilustrar os resultados encontrados com os algoritmos, sendo que isso poderia ser feito com qualquer um dos grafos.

Na Figura 4(b) encontram-se os resultados detalhados de todos os experimentos realizados com a instância bier127, nele fica evidente a superioridade do algoritmo

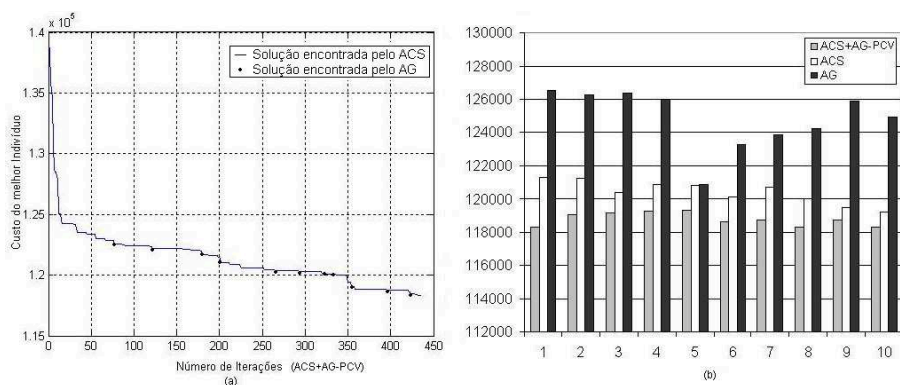


Figura 4: (a) gráfico de convergência obtido pelo ACS+AG-PCV para o grafo bier127 e (b) Gráfico de comparação do desempenho dos melhores resultados obtidos pelos algoritmos ACS+AG-PCV, ACS e AG para a instância bier127.

proposto ao longo dos experimentos, pois ele conseguiu encontrar melhores soluções que a dos outros dois algoritmos em todas as rodadas.

Na figura 5 estão os erros encontrados pelo algoritmo ACS+AG-PCV em relação a melhor solução encontrada, a média das soluções e a pior solução encontrada, em todos os grafos utilizados neste trabalho.

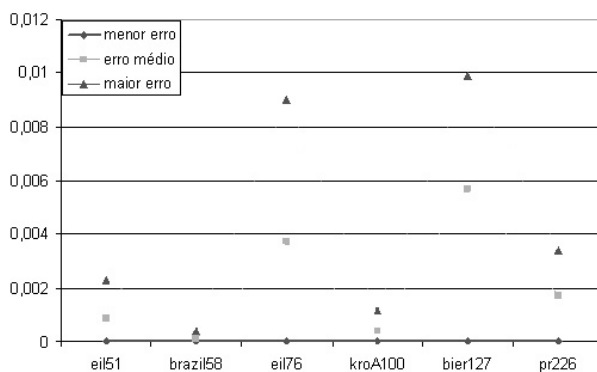


Figura 5: erros encontrados pelo algoritmo ACS+AG-PCV

Devido ao fato do Algoritmo Híbrido ACS+AG-PCV trabalhar com dois algoritmos, ele consome mais tempo de processamento que os algoritmos ACS e AG individualmente para realizar cada experimento (mediante a igualdade de parâmetros). Desta forma, a fim de obter uma comparação mais precisa, cada experimento do ACS e do AG foi testado utilizando o mesmo tempo de processamento consumido pelo ACS+AG-PCV.

Fazendo a análise dos resultados, podemos verificar na Tabela 1, que o algoritmo híbrido proposto foi superior em todos os experimentos realizados, encontrando solução ótima em todos os experimentos e média das soluções próximas do valor ótimo. Embora para algumas instâncias os algoritmos ACS (Tabela 1) e AG (Tabela 1) também tenham encontrado a solução ótima, as melhores médias das soluções foram as encontradas pelo algoritmo híbrido proposto. Note que os tempos de execução do algoritmo proposto foram bem pequenos. Desta forma os resultados obtidos neste trabalho demonstraram que os objetivos desejados foram satisfeitos.

Os erros apresentados na Figura 5 mostram resultados satisfatórios para os problemas utilizados, pois os erros médios são todos inferiores a 2%.

Para a implementação dos algoritmos e realização dos testes foi utilizada a ferramenta MATLAB 7.0 na seguinte plataforma: Processador Pentium IV - 3.0Ghz e 1024 Mb de RAM.

5. Conclusão

Este artigo apresenta uma nova Meta-heurística Híbrida que trabalha conjuntamente com os algoritmos ACS e AG para o PCV. Inicialmente, foram desenvolvidos todos os algoritmos necessários. Em seguida os algoritmos foram testados em diversas instâncias de tamanhos variados, de forma a mensurar o desempenho dos algoritmos. O Algoritmo Híbrido proposto mostrou ser uma técnica eficiente e ao mesmo tempo promissora para a resolução do PCV.

Em comparação com o desempenho obtido pelo algoritmo ACS+AG-PCV percebe-se que ele conseguiu obter melhores resultados, devido ao fato dele exercer um controle mais eficiente sobre a diversidade da população. Essa característica permitiu que o Algoritmo Híbrido obtivesse um melhor rendimento que os algoritmos ACS e AG individualmente.

Nos experimentos realizados, o Algoritmo Híbrido conseguiu encontrar soluções ótimas em todos os testes, enquanto que os algoritmos ACS e AG não conseguiram obter bons resultados com mesmo êxito.

Abstract. We present in this paper, a new hybrid meta-heuristic of Ant Colony Optimization (ACO) and Genetic Algorithms (GA) for to solve the Traveling Salesman Problem (TSP). The resolution of the Travelling Salesman Problem (TSP) becomes complex, therefore it involves a search in an enormous space of solutions that grows as increases the number of nodes of the graph. However the hybrid algorithm is proposed aiming at to get good results in way to skirt the question of the complexity.

Referências

- [1] M. Affenzeller, S. Wagner, A self-adaptive model for selective pressure handling within the of genetic algorithms. In: "Computer Aided Systems Theory", Eurocast, Spriger Verlag, 2809, pp. 384-393, 2003.
- [2] E. Bonabeau, M. Dorigo, G. Theraulaz, "Swarm Intelligence: From Natural to Artificial Systems", Oxford University Press, New York, 1999.

- [3] Bullnheimer, R.F. Hartz, C. Strauss, An improved ant system algorithm for the vehicle routing problem, *Annals of Operations Research*, 1999.
- [4] M. Dorigo, “Ottimizzazione, apprendimento automatico, ed algoritmi basati su metafora naturale (Optimization, Learning, and Natural Algorithms)”, Ph.D. Dissertation, Politecnico di Milano, Press, 1992.
- [5] M. Dorigo, L. M. Gambardella, Ant colonies for the traveling salesman problem, *BioSystems*, **43** (1997), 73-81.
- [6] M. Dorigo, T. Stützle, ACO Algorithms for the Traveling Salesman Problem, In *Evolutionary Algorithms in Engineering and Computer Science* (K. Miettinen, M. Makela, P. Neittaanmaki, J. Periaux, eds.) Wiley, 1999.
- [7] M.R. Garey, D.S. Johnson, “Computers and Intractability: a Guide to the Theory of NP-Completeness”, San Francisco, CA: W. H. Freeman, 1979.
- [8] J.H. Holland, “Adaptation in Natural and Artificial Systems”, 2 ed., MIT Press, MA, USA, 1992.
- [9] S. Lin, B.W. Kernighan, An effective heuristic algorithm for the traveling salesman problem. *Operations Research*, **21** (1973), 498-516.
- [10] P. Merz, B. Freisleber, Genetic Local Search for the TSP: New results. In: *IEEE International Conference on Evolutionary*, IEEE Press, pp. 159-164, Indianapolis, 1997.
- [11] Z. Michalewicz, “Genetic Algorithms + Data Structures = Evolution Programs”, Springer-Verlag, Berlin, 1996.
- [12] M.L. Pilat, T. White, Using genetic algorithms to optimize ACS-TSP, in “Proceedings of the Third International Workshop on Ant Algorithms - ANTS”, pp. 282-287, Springer-Verlag, 2002.
- [13] G. Reinelt, TSPLIB: A travelling salesman problem library. *ORSA Journal on Computing*, **3** (1991), 376-384.
- [14] S.C. Sari, H.D. Sherali, A. Bhootra, New tighter polynomial length formulations for the asymmetric traveling salesman problem with and without precedence constraints. *Operations Research Letters*, **33**, No. 1 (2007), 62-70.