

# AMAM: ARQUITETURA MULTIAGENTE PARA A SOLUÇÃO, VIA METAHEURÍSTICAS, DE PROBLEMAS DE OTIMIZAÇÃO COMBINATÓRIA

MARIA AMÉLIA L. SILVA\*, SÉRGIO R. DE SOUZA\*, HENRIQUE E. BORGES\*, SABRINA M. OLIVEIRA\*,  
ELIAS C. C. TEMPONI\*

*\*Programa de PósGraduação em Modelagem Matemática e Computacional  
Centro Federal de Educação Tecnologia de Minas Gerais  
Av. Amazonas 7675, Nova Gameleira - CEP 30510-000, Belo Horizonte, MG, Brasil*

Emails: [mamelials@gmail.com](mailto:mamelials@gmail.com), [sergio@dppg.cefetmg.br](mailto:sergio@dppg.cefetmg.br), [henrique@lsi.cefetmg.br](mailto:henrique@lsi.cefetmg.br),  
[oliveira.sabrina@gmail.com](mailto:oliveira.sabrina@gmail.com), [elias@uai.com.br](mailto:elias@uai.com.br)

**Abstract**— This article introduces the multiagent architecture AMAM (Multiagent Architecture for Metaheuristics). This architecture has, as objective, to combine several metaheuristics through the multiagent approach for solving Combinatorial Optimization Problems. Therefore, in the proposed architecture, each metaheuristic is developed as an autonomous agent which interacts in the environment in a cooperative way. The interaction between one or more agents arise through the exchange of messages on the search space of the problem and it searches for a better form of reaching the same objective. The AMAM is flexible enough to be used for solving different optimization problems, without the necessity of rewriting the algorithms. Two applications of the architecture for problems of different classes are used to show how the architecture can be used in the resolution of several optimization problems.

**Keywords**— Multiagent Architecture, Metaheuristics, Combinatorial Optimization

**Resumo**— Este artigo apresenta a arquitetura multiagente AMAM (Arquitetura MultiAgente para Metaheurísticas), que tem como objetivo combinar diversas metaheurísticas para a solução de problemas de otimização combinatória. Desta forma, na arquitetura proposta, cada metaheurística é projetada como uma agente autônomo e interage em seu ambiente (espaço de busca do problema) de forma cooperativa. A interação entre um ou mais agentes se dá através da troca de mensagens sobre o espaço de busca do problema e busca melhor atingir um mesmo objetivo. A AMAM é flexível o bastante para ser utilizada na solução de problemas de otimização de qualquer classe, sem que seja necessário reescrever os algoritmos. Duas aplicações da arquitetura foram desenvolvidas para demonstrar suas características. Os resultados mostram a viabilidade da proposta apresentada.

**Keywords**— Arquitetura Multiagente, Metaheurísticas, Otimização Combinatória.

## 1 Introdução

Problemas de otimização combinatória são encontrados em diversas áreas, *e.g.*, em aplicações industriais, processos logísticos, dentre outros. Na solução destes problemas, heurísticas e metaheurísticas têm recebido uma atenção especial dos pesquisadores. Isto porque, estes métodos, pertencentes à classe dos algoritmos aproximados (Milano and Roli, 2004), se, de um lado, não garantem a determinação da solução ótima dos problemas objeto de análise, de outro, permitem encontrar boas soluções viáveis em um tempo computacional reduzido. Assim, oferecem soluções eficientes para problemas de otimização, soluções estas construídas com base na combinação de diferentes estratégias e táticas de exploração do espaço de busca do problema proposto.

A atuação das metaheurísticas pode ser formulada como a de sistemas compostos por diversos agentes autônomo, em constante interação, que competem e colaboram entre si, para alcançar o objetivo comum. Recentemente, vários trabalhos têm utilizado a abordagem multiagente no tratamento de metaheurísticas e na solução de problemas de otimização.

Uma arquitetura agente para a solução do

Problema de Roteamento de Veículos (PRV) e suas variantes é proposta em (Thangiah et al., 2001). A arquitetura consiste de dois agentes principais: o Agente Leiloeiro, que é responsável pelo processo de leilão, nomeando clientes a veículos, e o Agente Veículo, que recebe as informações dos clientes, calcula o custo de inserção deste novo cliente em sua rota e envia sua oferta. Caso o Agente Leiloeiro nomeie o cliente ao veículo, este aceita o cliente em sua rota. Embora esta arquitetura seja aplicável às diversas variantes do PRV, seus agentes são altamente especializados, tendo suas habilidades limitadas ao tratamento de variáveis próprias desta classe de problemas.

Já na arquitetura MAGMA (*MultiAgent Metaheuristics Architecture*), apresentada em (Milano and Roli, 2004), uma metaheurística pode ser vista como o resultado da interação entre agentes. Organizada em níveis, a arquitetura MAGMA possui um ou mais agentes especializados atuando em cada um dos quatro níveis: Nível 0, que fornece uma solução factível para o nível superior; Nível 1, que melhora a solução; Nível 2, que tem a tarefa de guiar a busca para soluções mais promissoras; e Nível 3, que coordena todo o processo de busca. Entretanto, uma metaheurística, nesta proposta, não é um agente

isolado agindo autonomamente, mas surge da ação de diferentes agentes atuando nos vários níveis de abstração da arquitetura. O ambiente que os agentes percebem na arquitetura MAGMA é bem simples, sendo apenas o conjunto de agentes com quem eles se comunicam. Os agentes são definidos por tuplas  $T = \langle M, S \rangle$ , em que  $S$  é a estratégia usada pelo agente e  $M$  é o modelo do problema. Assim, todos os agentes, de todos os níveis da arquitetura, são “orientados a algoritmos”, ou seja, a definição do modelo  $M$  de cada agente é dependente do problema a ser tratado.

O presente artigo visa apresentar a Arquitetura MultiAgente para Metaheurísticas - AMAM. Nesta proposta, cada agente implementa uma certa metaheurística específica e tem autonomia de ação em sua exploração do espaço de busca do problema. Não obstante os agentes competirem entre si pelo alcance da melhor solução viável, eles, por outro lado, em sua ação, também colaboram entre si, trocando e compartilhando informações sobre seu estado e acerca do ambiente, *e.g.*, regiões promissoras do espaço de busca, regiões não recomendadas à exploração, soluções guias, soluções iniciais; parâmetros da busca, dentre outras questões.

A arquitetura AMAM foi construída de modo a ser flexível o bastante para permitir sua utilização em qualquer classe de problemas de otimização combinatória, e escalável, considerando que podem ser adicionados novos agentes, que implementam e encapsulam metaheurísticas específicas, com mínimo impacto no restante da arquitetura, ou seja, trata-se de uma arquitetura de alto nível de abstração. O ambiente, no qual os agentes metaheurísticos atuam, é parte integrante da arquitetura AMAM, devendo ser modelado para cada problema particular de otimização. A arquitetura também define e incorpora os mecanismos de comunicação entre os agentes e entre estes e seu ambiente. Assim, o esforço de modelagem para a criação de uma nova instância da AMAM para tratar diferentes problemas de otimização é minimizado.

Este artigo está organizado da seguinte forma: a seção 2 apresenta os principais aspectos da arquitetura AMAM, definindo-se os tipos de agentes bem como seu ambiente de atuação. Na seção 3, é apresentada a utilização da arquitetura em duas aplicações multiagentes envolvendo problemas de otimização combinatória de classes distintas. Finalmente, a seção 4 encerra este trabalho, apresentando algumas considerações finais e perspectivas de continuidade da construção dessa arquitetura multiagente.

## 2 Arquitetura AMAM

Esta seção descreve a arquitetura AMAM (Figura 1), definindo seus componentes (agentes

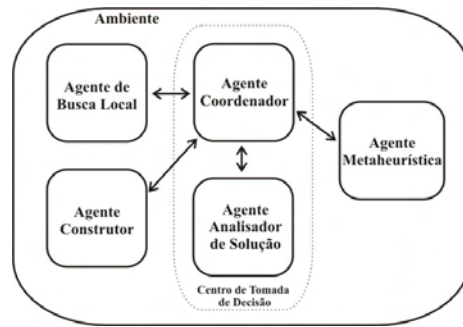


Figura 1: Arquitetura Multiagente AMAM

e ambiente), assim como as interações entre os agentes e entre estes e o ambiente. Os principais componentes dessa estrutura são: (1) Ambiente; (2) Agente Construtor; (3) Agente Busca Local; (4) Agente Metaheurísticas; e no centro de tomada de decisão, (5) Agente Coordenador e (6) Agente Analisador de Solução.

O principal objetivo da AMAM é a definição de uma estrutura geral (*framework*), à qual novos agentes (metaheurísticas) possam ser adicionados e diferentes problemas de otimização possam ser solucionados sem grande esforço adicional. Nas sub-seções seguintes, cada um dos elementos da arquitetura é melhor descrito.

### 2.1 Ambiente de Atuação

O ambiente é o elemento que possibilita a flexibilidade da arquitetura AMAM, pois corresponde ao espaço de busca do problema tratado, ou seja, todas as características próprias do problema, independente da forma como foi modelado, ficam definidas no ambiente. Desta forma, uma simples troca do ambiente da arquitetura corresponde à troca do problema a ser solucionado pelos agentes. Neste sentido, o esforço computacional de solução de diferentes problemas é reduzido à modelagem dos seus atributos específicos.

A forma geral do ambiente, apresentada na Figura 2, define o modelo básico de um problema de otimização combinatória. Esta forma é definida considerando-se que, na modelagem de um problema de otimização, é necessário definir-se a forma como a solução do problema será representada e, além disso, que todas as informações essenciais para a busca de soluções sejam conhecidas.

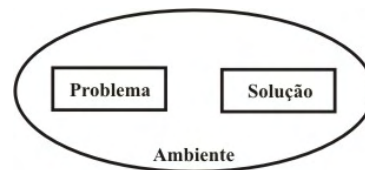


Figura 2: Modelo do Ambiente

O agente percebe as características do problema e age construindo soluções e explorando o es-

paço de busca. As funções de vizinhança são uma das possibilidades de interação dos agentes com o ambiente na arquitetura multiagente proposta, possibilitando a construção de novas soluções a partir da modificação de soluções já existentes. Como as funções de vizinhança também são específicas para cada modelo de problema, precisam ser definidas como atributos da solução. Desta forma, a definição dos diferentes tipos de vizinhança, assim como das formas de adição de elementos utilizados pelas heurísticas construtivas, podem ser especificadas na estrutura Solução do modelo, apresentado na Figura 2.

## 2.2 Agentes de Busca

Para uma melhor compreensão da arquitetura AMAM, é importante definir o papel que cada agente desempenha. Os Agentes Construtor, Busca Local e Metaheurística são classificados como agentes de busca, por possuírem, como principal função, a busca de soluções no espaço de busca do problema. Eles são descritos a seguir.

**Agente Construtor de Solução:** Agentes formados por heurísticas construtivas ((Blum and Roli, 2003)). São responsáveis pela construção de soluções iniciais, que serão utilizadas pelos Agentes Busca Local e Metaheurísticas.

**Agente Busca Local:** Agentes responsáveis por refinar soluções já encontradas, a partir de algoritmos de Busca Local ((Blum and Roli, 2003)).

**Agente Metaheurística:** Agentes que utilizam algoritmos de metaheurísticas para encontrar “boas” soluções para problemas de otimização combinatória. Estes algoritmos, normalmente, utilizam heurísticas subordinadas (busca local) e estratégias que permitem escapar de ótimos locais para a exploração do espaço de busca.

## 2.3 Agentes do Centro de Tomada de Decisão

As várias formas que definem a organização de uma sociedade de agentes permitem determinar estruturas hierárquicas e diretrizes para que os agentes possam atingir um objetivo comum em cooperação. Nesta proposta, um agente coordenador central é utilizado para intermediar a comunicação entre os demais agentes, caracterizando uma relação de gerência e reduzindo a complexidade de comunicação.

**Agente Coordenador:** tem a função de coordenar as atividades e a comunicação entre os agentes em um ambiente multiagente. Em seu trabalho de coordenação, este agente tem, além de realizar trocas de informações, também a responsabilidade por realizar negociações, acordos e tradução das mensagens entre os diferentes agentes. A cooperação entre os agentes surge por meio do envio de informações como regiões promissoras do espaço de busca, regiões não recomendadas à exploração, soluções guias, soluções

iniciais, parâmetros da busca. A comunicação entre os agentes acontece através da troca de mensagens. Desta forma, o agente coordenador conhece todos os outros agentes da arquitetura, e é responsável, quando recebe uma mensagem com alguma solicitação de serviço, por encaminhá-la para o agente que pode realizar a operação solicitada. Outros tipos de comunicação também podem ser inseridos, como, por exemplo, a arquitetura de quadro-negro, que, através de um repositório de dados, realiza a troca de informações (e.g. soluções) entre os agentes.

**Agente Analisador de Soluções:** responsável pela análise e tomada de decisão com relação às soluções obtidas pelos Agentes Metaheurísticos, selecionando, dentre elas, a melhor solução.

## 2.4 Estrutura Interna dos Agentes de Busca

Tanto os agentes construtores e de busca local quanto os agentes metaheurísticos possuem uma mesma estrutura interna genérica, diferindo entre si apenas na implementação. Esta estrutura genérica é baseada na arquitetura geral de um agente proposta em (Resende, 2003). Assim, fica bastante simplificado o projeto de novos agentes de busca. A figura 3 apresenta um diagrama ilustrativo dos principais componentes internos desta estrutura, que são descritos abaixo.

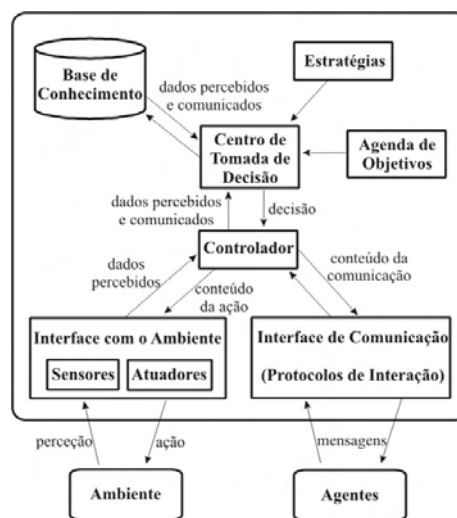


Figura 3: Estrutura Interna dos Agentes de Busca

**Base de Conhecimento:** componente que mantém informações acerca do ambiente, informações necessárias para a comunicação entre os agentes, informações sobre os demais agentes, bem como o histórico de busca do próprio agente. Mostra a representação do ambiente;

**Interface de Comunicação:** define os protocolos de comunicação entre os agentes, permitindo a troca de mensagens entre eles.

**Interface com o Ambiente:** envolve os sensores e atuadores do agente. Os sensores possibilitam ao

agente perceber o ambiente, enquanto os atuadores proporcionam ao agente a capacidade de agir e modificar seu ambiente;

**Centro de Tomada de Decisão:** componente responsável por tomar decisões com base nos conhecimentos obtidos, nos objetivos e nas estratégias do agente. Ele deve definir qual o melhor caminho para a busca do agente;

**Agenda de Objetivos:** lista de objetivos e de restrições a serem satisfeitas, obtida a partir do ambiente (*i.e.*, do problema particular de otimização combinatória a ser resolvido);

**Estratégias:** encapsula a estratégia de busca do agente, ou seja, o algoritmo da heurística ou metaheurística implementada. Este componente é responsável por encontrar soluções para o problema representado pelo ambiente. Desta forma, possibilita o isolamento do problema e do método utilizado e permite que esta estrutura geral possa ser usada na definição de qualquer agente de busca;

**Controlador:** intermedia e coordena todo o processo interno do agente, ou seja, as trocas de informação entre seus componentes internos.

Em linhas gerais, o *modus operandi* de um certo agente de busca pode ser resumido assim: por meio dos sensores, em sua interface com o ambiente, o agente captura as características do ambiente (dados necessários para a solução do problema em questão), enquanto os seus atuadores constroem e modificam as soluções (ambiente). Neste processo, o agente pode se comunicar com outros agentes, via sua interface de comunicação. Assim, pode obter a colaboração destes, através da troca de informações úteis para a busca. Tanto os dados percebidos do ambiente, quanto aqueles comunicados por outros agentes, são usados para auxiliar a tomada de decisão do agente. Durante todo o processo de busca, o centro de tomada de decisão realiza escolhas, visando minimizar (ou maximizar) a função de custo do problema.

### 3 Especializações da Arquitetura AMAM

A arquitetura AMAM foi especializada em várias instâncias, com os seguintes objetivos: (1) demonstrar o correto funcionamento da arquitetura; (2) demonstrar a flexibilidade da arquitetura em resolver diferentes problemas de otimização, sem que mudanças significativas sejam feitas; e (3) demonstrar que a adição de novos agentes melhora o desempenho da arquitetura como um todo. Deve-se ressaltar que não é objetivo deste artigo superar as melhores soluções encontradas na literatura para problemas em análise, mas, sim, demonstrar as características e benefícios obtidos a partir da perspectiva multiagente, tais como flexibilidade, escalabilidade e encapsulamento.

#### 3.1 Arquitetura AMAM para o PRVJT

Uma especialização da arquitetura AMAM foi desenvolvida para tratar o Problema de Roteamento de Veículos com Janela de Tempo (PRVJT).

No Problema de Roteamento de Veículos (PRV), o interesse é a determinação de rotas com o menor custo, de modo a satisfazer um conjunto de clientes espalhados geograficamente. O PRVJT é um caso especial do PRV clássico, em que os clientes podem ter janelas de tempo para atendimento associadas a eles. Uma descrição completa destes problemas pode ser encontrada em (Toth and Vigo, 2002).

A especialização da arquitetura para o PRVJT (figura 4) utiliza o Agente Metaheurística *Variable Neighborhood Search* (VNS) (Hasen et al., 2003), em conjunto com o Agente Construtor, implementando a heurística construtiva Gulosa (*Greedy Heuristics*) (Blum and Roli, 2003) e o Agente Busca Local implementando o Método Primeiro de Melhora (*First Improvement Method*) (Blum and Roli, 2003).

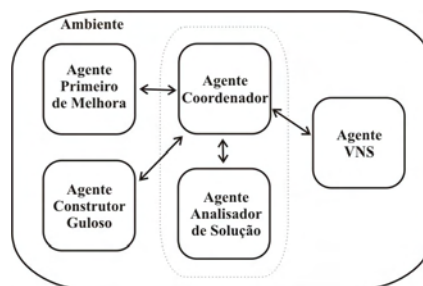


Figura 4: Arquitetura AMAM para o PRVJT e o PCBG

Para este caso, o ambiente modelado para o PRVJT é apresentado na Figura 5, onde o problema possui todas as informações dos clientes (localização, janelas de tempo, demanda, dentre outras) e a solução é representada por um conjunto de rotas.



Figura 5: Modelo do Ambiente do PRVJT

Experimentos utilizando a arquitetura AMAM para o PRVJT foram realizados e o comportamento médio obtido é apresentado na Tabela 1. Foram utilizadas as instâncias de classe C100, apresentadas em (Solomon, 1987).

#### 3.2 Arquitetura AMAM para o PCBG

Esta sessão busca demonstrar o caráter genérico da arquitetura, ou seja, como uma especializa-

Tabela 1: Resultados para o PRVJT.

Prob.	Média	Melhor	Prob.	Média	Melhor
C101	2527,00	827,30	C201	1032,20	591,56
C102	2379,60	827,30	C202	1014,40	591,56
C103	2334,10	827,30	C203	983,90	591,56
C104	2310,40	827,30	C204	1001,60	590,60
C105	2511,20	827,30	C205	988,40	586,40
C106	2368,60	827,30	C206	1032,20	586,40
C107	2409,90	827,30	C207	1032,20	585,80
C108	2409,90	827,30	C208	1046,30	585,80
C109	2482,10	827,30			

ção da arquitetura para um determinado problema pode ser facilmente utilizada na solução de diferentes classes de problemas, pela simples troca de ambientes. Desta forma, na especialização para o Problema de Corte Bidimensional Guilhotinado (PCBG), foram utilizados os mesmos agentes da especialização para o PRVJT, como mostra a Figura 4. A única modificação, em relação à aplicação para o PRVJT, se refere ao modelo do ambiente utilizado, conforme ilustra a Figura 6.

A classe de problemas associada ao PCBG envolve a busca de soluções as mais econômicas para o corte de grandes objetos, visando à produção de objetos menores, de dimensão e demanda específicas. No caso em análise, os objetos a serem cortados possuem forma retangular e têm, apenas, duas dimensões relevantes. Uma revisão dessa classe de problemas pode ser encontrada em (Morabito and Arenales, 1992).



Figura 6: Modelo do Ambiente do PCBG

Os problemas-teste usados para os experimentos realizados com o PCBG são: (1) 3 itens com dimensões de 30x30, 15x15, 10x10, com demandas a serem atendidas de 25, 20, 30, respectivamente; (2) 4 itens com dimensões de 19x13, 27x15, 35x20, 15x10, com demandas a serem atendidas de 25, 19, 16, 30, respectivamente; e (3) 5 itens com dimensões de 17x10, 27x15, 35x29, 20x15, 37x23, com demandas a serem atendidas de 15, 14, 7, 12, 6, respectivamente. A partir dos testes realizados, foi obtido o comportamento médio para cada um destes problemas-teste, como mostra a Tabela 2.

### 3.3 Escala Efetiva

O objetivo aqui é demonstrar que a arquitetura AMAM permite a fácil adição de novos agentes e que seu desempenho pode ser melhorado através

Tabela 2: Resultados para o PCBG.

Prob.	Média	Tempo
1	21012	17,7
2	27775	39,2
3	24738	32,6

destas adições. Talukdar (Talukdar et al., 1992) apresenta um conceito relacionado ao efeito de escala de uma organização baseada em computador. Assim, uma organização é de escala-efetiva (*scale-effective*) se seu desempenho melhora com o tamanho, ou seja, se existem agentes, tais que sua adição melhora a qualidade das soluções encontradas, e computadores, cuja adição melhora a velocidade com que esta solução de qualidade é obtida. Desta forma, em uma organização escala-efetiva, é minimizado o esforço de adição de novos componentes. Entretanto, em contraponto, uma organização que não possui esta característica enfrenta problemas para adicionar e modificar suas partes, tornando uma simples alteração uma situação trabalhosa, com um grande impacto negativo sobre o desempenho global.

A partir deste conceito, duas especializações da arquitetura AMAM foram implementadas para discutir esta característica. A primeira (Figura 7) utiliza dois Agentes Metaheurísticas, implementando os métodos *Iterated Local Search* (ILS) e VNS, para a solução do PRVJT.

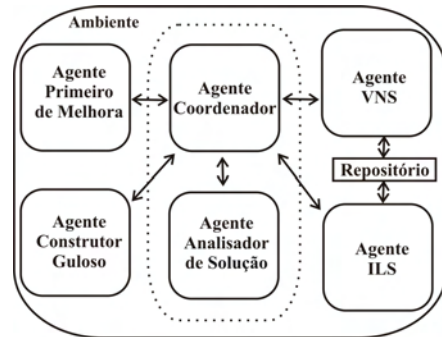


Figura 7: Arquitetura AMAM utilizando os Agentes Metaheurísticas VNS e ILS.

A segunda (Figura 8) acrescenta à primeira um novo Agente Metaheurística, implementando o método GRASP (*Greedy Randomized Adaptive Search Procedure*) (Feo e Resende (1995)), sendo utilizada também na solução do PRVJT.

O modelo do ambiente utilizado nas duas especializações, correspondente ao problema PRVJT, é o mesmo apresentado na figura 5. Ambas as especializações foram desenvolvidas utilizando-se o sistema de comunicação chamado quadro-negro (blackboard) (Ferreira (2002)). Nesta abordagem, uma espécie de repositório (quadro-negro), onde os agentes podem ler e escrever mensagens, é usado como meio

de interação. Este repositório, no presente artigo, tem a formulação de uma lista tabu, de modo que soluções não recomendadas são colocadas para que possam ser evitadas pelos agentes. Desta forma, o ambiente é utilizado também como intermediador de comunicação, já que, através do repositório, os agentes podem inserir e retirar soluções.

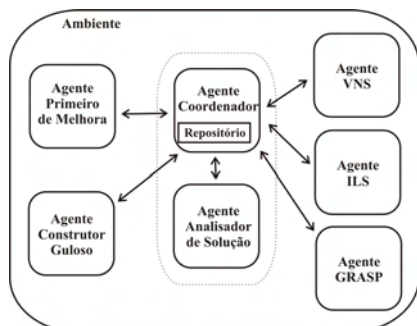


Figura 8: Arquitetura AMAM utilizando os Agentes Metaheurísticas VNS, ILS e GRASP.

Para uma avaliação experimental, as duas especializações foram testadas, utilizando-se a instância C101 (Solomon, 1987) com 100 consumidores, e obteve-se o comportamento médio apresentado na figura 9. Observa-se, então, que, com a adição do terceiro agente, foi possível alcançar uma significativa melhora no valor da função objetivo da solução. Da mesma forma, pode-se afirmar que existe uma tendência de diminuição do valor da função objetivo em relação ao crescimento do número de iterações; não obstante, as metaheurísticas utilizadas se prenderem a ótimos locais, justifica-se a presença de picos no gráfico.

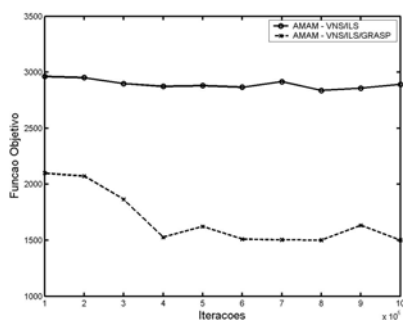


Figura 9: Comparação entre as especializações da arquitetura: AMAM-VNS/ILS e AMAM-VNS/ILS/GRASP.

#### 4 Conclusões

Neste artigo, é apresentada uma arquitetura multiagente para a solução de problemas de otimização combinatoria, através do uso de metaheurísticas. Nesta proposta, as metaheurísticas são definidas como agentes que atuam na exploração

do espaço de busca. Além disso, os agentes possuem capacidades de comunicação entre si e com o seu ambiente, possibilitando um comportamento cooperativo que tende a atingir um objetivo comum, qual seja, encontrar uma “boa” solução. A arquitetura AMAM define um *framework*, que permite a comparação do desempenho das diversas metaheurísticas na solução de um determinado problema.

Desta forma, a arquitetura AMAM se mostra flexível o bastante para solucionar diferentes problemas de otimização, sem grande impacto no restante da arquitetura. A adição de novos agentes (metaheurísticas) também é facilitada, já que a arquitetura define um *framework* para metaheurísticas híbridas. As especializações apresentadas permitem a avaliação experimental da arquitetura proposta, demonstrando como estas características estão presentes.

#### Referências

Blum, C. and Roli, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison, *ACM Computing Surveys* **35**(3): 268–308.

Hasen, P., Mladenovic, N. and Pérez, J. (2003). Variable neighbourhood search, *Revista Iberoamericana de Inteligencia Artificial* (19): 77–92.

Milano, M. and Roli, A. (2004). Magma: a multi-agent architecture for metaheuristics, *IEEE Transaction on Systems Man and Cybernetics* **34**(2).

Morabito, R. and Arenales, M. (1992). Um exame dos problemas de corte e empacotamento, *Pesquisa Operacional* **1**(12).

Resende, S. (2003). *Sistemas Inteligentes - Fundamentos e Aplicações*, Manole, Porto, Portugal.

Solomon, M. M. (1987). Algorithms for vehicle routing and scheduling problems with time windows constraints, *European Journal of Operational Research* **35**: 254–266.

Talukdar, S., Baerentzen, L., Gove, A. and Souza, P. (1992). Asynchronous teams: Cooperation schemes for autonomous agents, *Journal of Heuristics* **4**.

Thangiah, S., Shmygelska, O. and Mennell, W. (2001). An agente architecture for vehicle routing problems, *In Proceedings of the 2001 ACM Symposium on Applied Computing* pp. 517–521.

Toth, P. and Vigo, D. (2002). *The Vehicle Routing Problem*, SIAM - Society for Industrial and Applied Mathematics, Philadelphia.